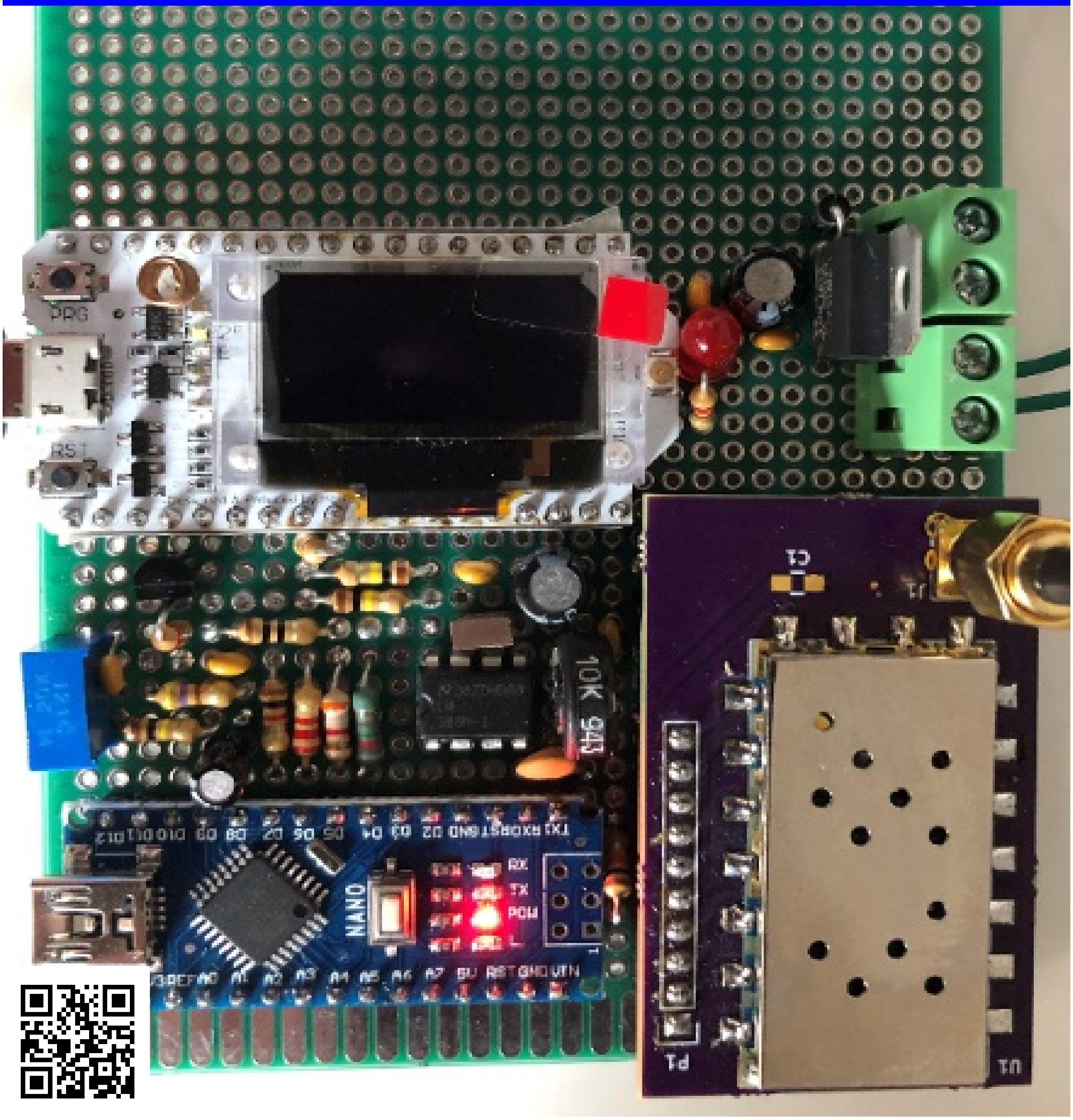
RAZzies

Maandblad van de Radio Amateurs Zoetermeer

Juni 2019

Met in dit nummer:

- Experimentjes met weerstanden
- VK3YE Micro 40 DSB Transceiver
- Opa Vonk: Cursus Kicad - deel 1
- APRS iGate met Arduino (B)
- Aanvulling 2m antenneversterker
- Chinese componenten
- Afdelingsnieuws



Colofon

RAZzies is een uitgave van de Radio Amateurs Zoetermeer. Bijeenkomsten van de Radio Amateurs Zoetermeer vinden plaats op elke tweede en vierde woensdag van de maanden september - juni om 20:00 uur in het clubhuis van de Midgetgolfclub Zoetermeer in het Vernède sportpark in Zoetermeer.

Website:

http://www.pi4raz.nl

Redactie:

Frank Waarsenburg
PA3CNO
pa3cno@pi4raz.nl

Eindredactie:

Robert de Kok
PA2RDK
pa2rdk@pi4raz.nl

Informatie:

info@pi4raz.nl

Kopij en op- of aanmerkingen kunnen verstuurd worden naar razzies@pi4raz.nl

Nieuwsbrief:

http://pi4raz.nl/maillist/subscribe.php

Van de redactie

Een superdikke RAZzie deze keer. En het is nog niet eens zomer (wie kent nog het dubbeldikke zomernummer van Elektuur HI). Er was gewoon een hoop te melden deze keer. Niet in de laatste plaats de iGate. We hebben het er al vaak over gehad, maar in deze RAZzie vind je dan het hele artikel over deze ontwikkeling. Met dit relatief goedkope ontwerp zet je een volledig functionerende APRS gateway naar het internet neer. Geen computer of 2m set meer nodig: alleen WiFi is voldoende om ontvangen APRS packets aan te bieden bij de website van aprs.fl. Dat zou amateurs moeten stimuleren om het APRS netwerk (weer) te gaan ondersteunen. En verder liep ik tegen een leuk ontwerp aan van een dubbelzijband 40m transceiver. Dubbelzijband heeft het voordeel dat de opbouw van zo'n transceiver vrij eenvoudig kan zijn, zonder veel af te doen aan de prestaties. En vergis je niet, je hebt niet veel vermogen nodig om een verbinding te maken. Nou neemt de moeite die je moet doen voor het maken van een verbinding wel toe naarmate het vermogen afneemt, maar de voldoening als een verbinding dan slaagt neemt ook toe en dat is waar het in een hobby toch om draait nietwaar. Het ontwerp biedt volop mogelijkheden om te exmerimenteren en uit te breiden en ik kan het de beginnende amateur dan ook van harte aanbevelen om met dit ontwerp zelf aan de slag te gaan.

Experimentjes met weerstanden

Frank Vermeulen PA5FJM

Aangezien fouten zoeken niet mijn sterkste kant is, lijkt het me goed om experimentjes te gaan doen. Uiteindelijk schijnen we het meeste te leren van de dingen die fout gaan, hi.

Voor de experimentjes neem ik het boek “Electronica, echt niet moeilijk”. Bestaande uit 3 delen: gelijkstroom, wisselstroom en digitale techniek. Digitale techniek weet ik nog niet of ik die ga doen. Eerst uitgegeven als 3 afzonderlijke boekjes, later gebundeld in 1 boek. Wat hebben we aan onderdeeltjes nodig? Lijstje gemaakt van het deel over gelijkstroom en terug de kelderbox in om alles bij elkaar te zoeken. Wat bleek? Ik had zelfs nog een BF245 en een LDR. Jippie. Ik had bijna alles. Ik kon namelijk geen weerstandjes vinden van 470 kOhm. Hmmm, hoe gaan we dat oplossen? Ik vond nog wel weerstandjes van 1 MOhm. Aha: parallel schakelen van gelijke weerstanden halveert de waarde. Ergo, dan kom ik uit op 500k. Dat is bijna 470 kOhm. Kortom, ik had weerstandjes van 470 kOhm, hi. Alles in een afsluitbaar bakje en bijna klaar voor de start.

Voeding

Zo goed als alle schakelingetjes in

bovengenoemd boek zijn met 4.5V ontworpen. Het was wel verleidelijk om een voeding, al was het maar een eenvoudige, te gaan maken, maar ik had me voorgenomen om het stap voor stap te (gaan) doen, dus we doen het wel met een batterij.



Zouden 4.5V batterijen nog wel bestaan? In de lokale supermarkten had ik ze al een aantal jaren niet meer gezien. Een mini-zoektocht op internet bood genoeg mogelijkheden. Zouden de bouwmarkten ze ook nog hebben? Kijken op internet leverde een bouwmarktketen in België op die ze ook nog zou kunnen hebben. Mooi, er zit er eentje een goed kwartiertje fietsen van mij vandaan (ik woon op ongeveer 100 meter van de Belgische grens). Op een zonnige middag de fiets gepakt en naar de bouwmarkt in Kapellen in België gefietst en ja hoor, er hingen nog 4.5V batterijen aan het rek. De voeding was dus ook geregeld.

Weerstanden

Weerstanden zijn eigenlijk nog de makkelijkste onderdelen en dit was waar het boek ook mee begint en eigenlijk geloofde ik het wel. Maar toch, het is de basis en natuurlijk toch ook een goede oefening om de wet van Ohm (U = I*R) te oefenen. Daarnaast is het ook een goede oefening om vermogensberekeningen (de bekende formule P = U*I) te berekenen.

Dus, eerder genoemd boek erbij gepakt en beginnen met het eerste schakelingetje. Een serieschakeling van 3 weerstandjes. Gewoon lekker simpel met enkele draadjes met krokodillenklemmetjes. Zie fig. 3-1. De berekeningen komen redelijk overeen met wat ik heb gemeten. Kleine verschilletjes, maar dat kan natuurlijk te maken hebben met de tolerantie van de weerstanden en misschien dat mijn meter toch niet helemaal in orde meer is. Maar op dit moment gaat het ook meer om het idee. En ach, 0.53V berekend en 0.59V gemeten voor R1 vind ik zo slecht nog niet. Verschil tussen theorie (berekeningen) en praktijk (metingen), HI.

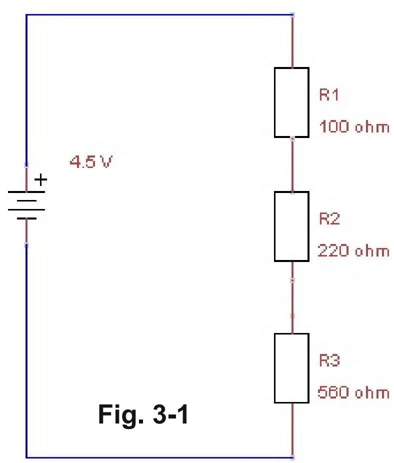


Fig. 3-1

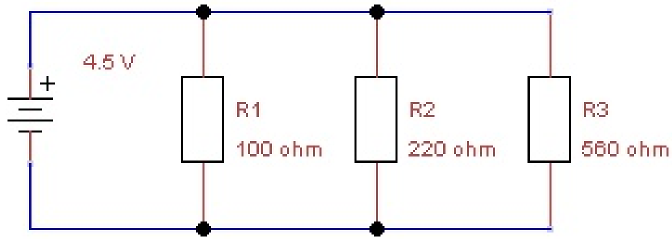


Fig. 3-2

Hetzelfde gedaan voor de parallelschakeling. Zie fig. 3-2. Uiteraard dan voor de stroom, want de spanning is voor alle 3 de weerstandjes natuurlijk gelijk. Echter, hier liep ik toch tegen een probleem aan. Volgens de multimeter was er geen stroom te bekennen. Snappen deed ik het niet, want ik kon wel spanning meten. Schakelingetje opnieuw opgebouwd, nog steeds geen stroom. Dat begint al goed. Zo'n simpel schakelingetje en dat werkt al niet…… Alles nog eens gecontroleerd: ja hoor, meter in serie geschakeld met de weerstanden, weerstanden zelf goed aangesloten, het zou toch moeten werken. Maar, nog steeds geen stroom. Nog maar eens controleren. Nu letterlijk alle draden met een wijsvinger volgen vanaf de plus van de batterij naar de min van de batterij. Ehm, wil ik stroom meten, moet ik natuurlijk wel de meetsnoeren in de juiste aansluiting op de multimeter doen. Voor stroommeting zit er namelijk een aparte aansluiting op de multimeter. Ik kan de schakelaar wel op stroommeting zetten, maar als ik de meetsnoeren in de aansluiting voor spanning laat zitten, zal hij nooit stroom meten. Ik geef toe dat ik toch wel enige tijd met het schaamrood op mijn wangen heb gezeten. Afijn, snoer in de goede aansluiting gedaan en zie daar, ik kon stroom meten, HI. Ook hier weer enkele kleine verschilletjes tussen wat ik heb gemeten en berekend. Aan de andere kant: 7.8

mA gemeten en 8 mA berekend voor R3 is toch ook niet slecht.

Brug van Wheatstone

Bezig met weerstanden, betekent natuurlijk ook dat de brug van Wheatstone om de hoek komt kijken. Het is natuurlijk leuk wat er beweerd wordt over de brug van Wheatstone, maar is dat ook zo?

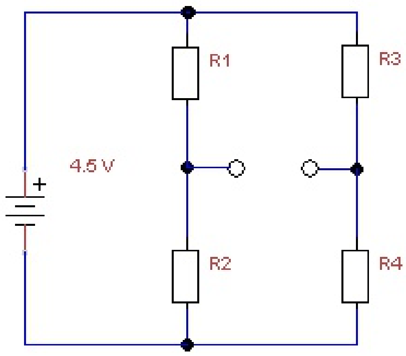


Fig. 3-3

Een brugje opgebouwd met weerstandjes van 1MOhm, zie fig. 3-3, en eens kijken wat de spanning doet. Is toch weer een spannend moment. Zitten mijn meetsnoeren in de juiste aansluitingen op de multimeter? Check. Dan gaat dat alvast niet fout, HI. In beide takken inderdaad, zoals verwacht, de helft van de batterijspanning gemeten over R2 en R4. Dus theoretisch zou het spanningsverschil tussen beide takken 0V moeten zijn. En ja hoor, het klopt nog ook, HI: Tussen de knooppunten van R1/R2 en R3/R4 is een spanningsverschil van 0V. En volgens mij behoor ik officieel te zeggen: een potentiaalverschil van 0V.

Saillant detail vind ik wel dat Charles Wheatstone nog geeneens de uitvinder geweest schijnt te zijn van de brug van Wheatstone, maar ene Samuel Hunter Christie ongeveer 10 jaar eerder. Ik vind het dan ook interessant om te kijken wie was die Wheatstone? Charles schijnt het idee van Samuel alleen maar verbeterd te hebben. Althans, volgens Wikipedia.

Voor de gein heb ik een tak vervangen door een potmeter, zie fig. 3-4. Ik had nog een potmeter van 5 kOhm liggen en deze maar gebruikt. R1 en R2 bleven 1 MOhm. Toch grappig om dan te zien dat inderdaad een potentiaalverschil tussen knooppunt R1/R2 en de loper van de potmeter 0V is bij het midden van de potmeter. En dat, als je de loper verdraait van de ene uiterste naar de andere uiterste stand, de spanning netjes verloopt van +2.3V naar -2.3V.

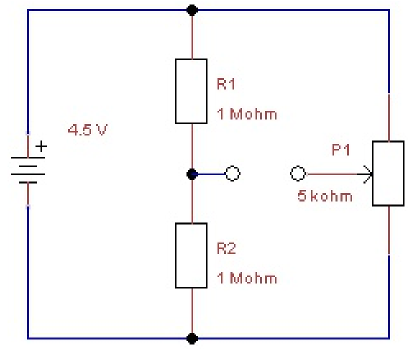


Fig. 3-4

Is natuurlijk simpel te verklaren, want een potmeter is eigenlijk niets anders dan een spanningsdeler, waarbij de stand van de loper de weerstand bepaalt. En aangezien ik het midden van de 2 losse weerstanden als nulpunt neem, wordt de spanning in de uiterste standen van de loper van de potmeter, gehalveerd. Gehalveerd, omdat de 2 losse weerstanden dezelfde waarde hebben (er natuurlijk vanuit gaande dat er geen tolerantie aanwezig is, ofwel het ideale geval).

Met deze schakeling kreeg ik nog een ander ideetje. Bij een gelijkspanning gaat de spanning van +2.3V naar -2.3V. Zie ik hier iets van terug bij een blok- of sinusspanning? Ik loop dan eigenlijk wel vooruit op mijn eigen planning, want wisselspanning(en) zou ik pas later gaan doen, maar toch wint de nieuwsgierigheid, HI.

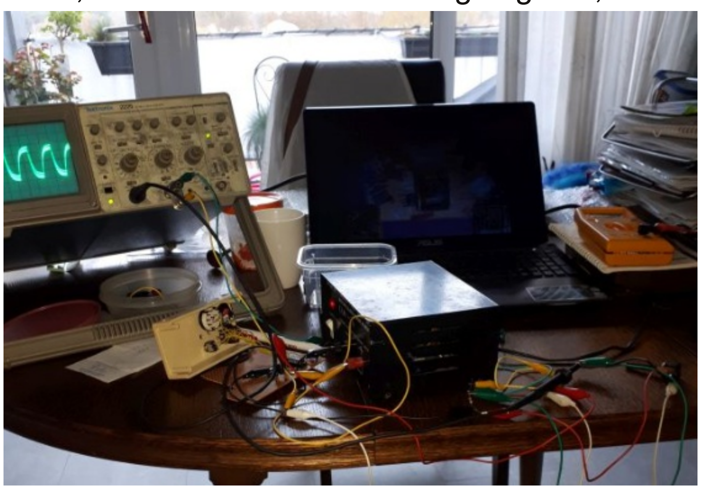


Fig. 3-5

De laatste schakeling nogmaals opgebouwd, maar in plaats van de 4.5V batterij had ik een zelfgemaakt functiegeneratortje aangesloten. In fig. 3-5 is enigszins te zien hoe ik heb zitten werken. Zoals te zien is het niet echt ideaal, maar het gaat dan ook om het idee. Tijdens het

uitproberen kreeg ik het helaas niet voor elkaar om met 2 kanalen op de oscilloscoop te werken. Hier moet ik dus nog eens een keertje een avondje voor gaan zitten. Helaas moeten we het nu dus maar even met 1 kanaal/signaal doen.

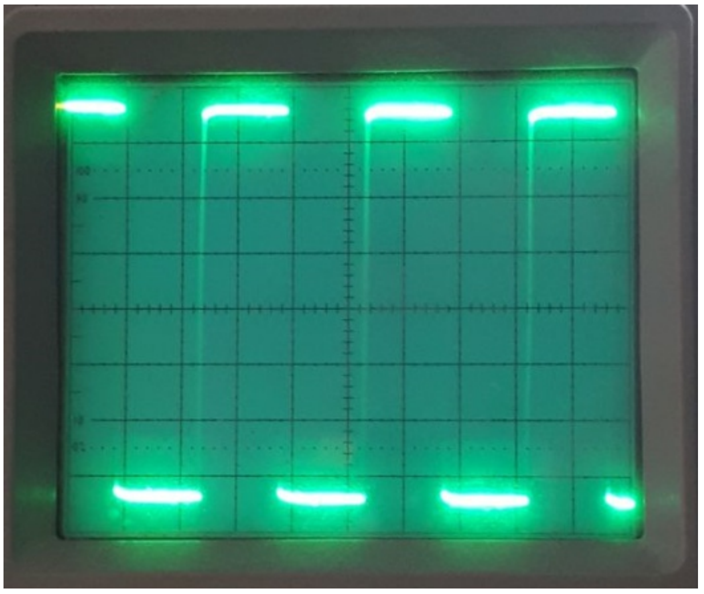


Fig. 3-6

Arbitrair heb ik er voor gekozen om de blokgolf te gebruiken (zie fig. 3-6), ervan uitgaande dat voor een sinus hetzelfde zal gelden. Een blokgolf is per slot van rekening samengesteld uit een heleboel sinussen. Ik weet dat ik nu even heel kort door de bocht ga, want bij een transistorschakeling kan ik me voorstellen dat deze theorie (een blokgolf en sinus geven hetzelfde effect) niet opgaat. Leuk idee om straks eens uit te proberen, bij zowel een diode als transistor (en FETs).

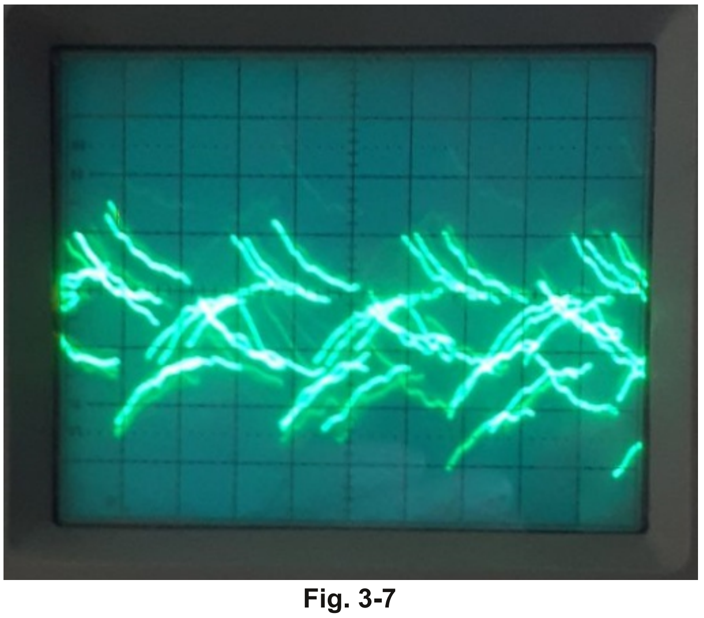


Fig. 3-7

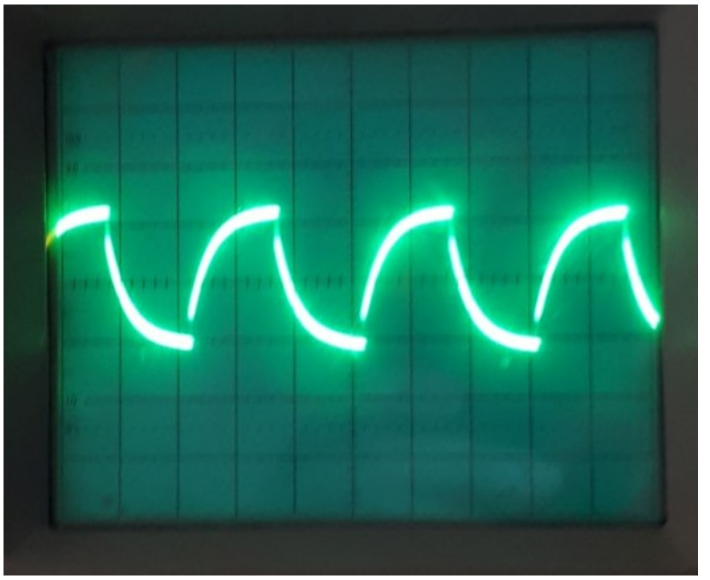


Fig. 3-8

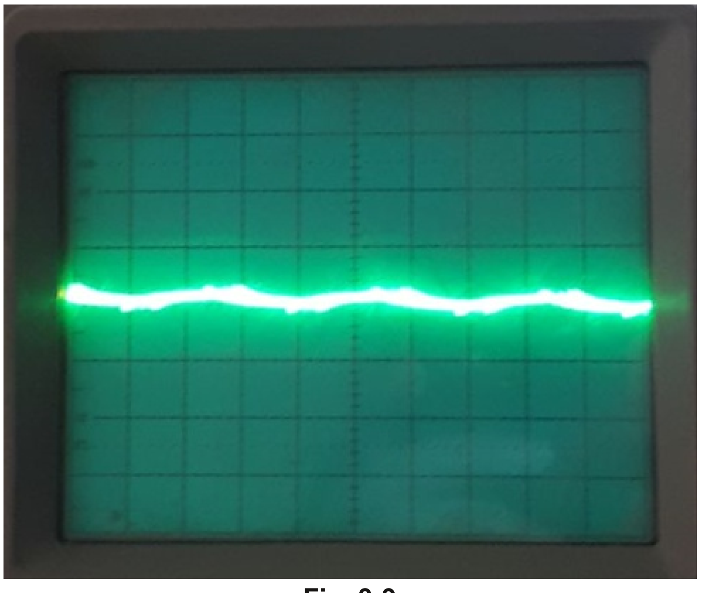


Fig. 3-9

Ik begon met de potmeter in de ene uiterste stand, en wel zodanig dat de maximale weerstand op de loper stond. Bij gelijkspanning dus +2.3V. Er kwam in eerste instantie een belabberd beeld uit (zie fig. 3-7), maar na de massaverbinding wat verbeterd te hebben kreeg ik al een acceptabeler signaal op de oscilloscoop te zien. Dit is te zien in fig. 3-8. Vervolgens aan de potmeter gaan draaien en langzaam verkleinde de amplitude van de blokgolf naar fig. 3-9. Min of meer geen verschil tussen de beide takken van de weerstanden, net zoals bij gelijkspanning. Daar was het spanningsverschil tussen de twee takken ook 0V. Vervolgens de potmeter doorgedraaid naar de andere uiterste stand, die van -2.3V bij gelijkspanning. En het grappige was (althans, dat vond ik toch) dat ik

op de oscilloscoop hetzelfde plaatje (zie fig. 3-10) kreeg te zien als in de andere uiterste stand van de potmeter. Kennelijk maakt het bij een blokgolf niet uit in welke uiterste stand de potmeter staat. (*Dat doet het wel, de fase is 180 graden gedraaid. Alleen zie je dat niet omdat de triggering van de oscilloscoop dat corrigeert -red.*)

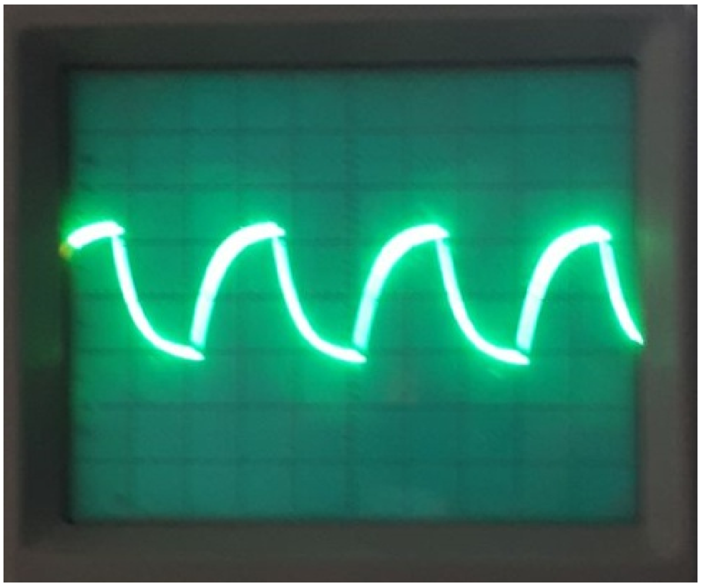


Fig. 3-10

Er vielen me 3 dingen op:

1. De amplitude van de blokgolf was in de uiterste standen meer dan gehalveerd t.o.v. de originele amplitude. Ik snap dat de amplitude lager is in de brug, want ik neem het nulpunt tussen 2 dezelfde weerstanden, dus net als bij gelijkspanning, maar dat de amplitude meer dan gehalveerd is, nee, dat snap ik niet. Maar goed, ik heb de theorie van de wisselspanning/stroom nog niet opnieuw bestudeerd. *(De reden is vermoedelijk dat de probe een 1M probe is, tenzij er een 10x stand op zit: die kan 10M zijn. Maar doordat Frank 1M weerstanden in de brug gebruikt, vindt er een extra spanningsdeling plaats. Dat zou de reden voor het verval van de amplitude kunnen zijn -red.)*

2. De blokgolf was geen blokgolf meer, maar de opgaande flank werd afgerond. Dit was tegen mijn verwachting: ik had gewoon een normale blokgolf verwacht. Wellicht dat het te maken heeft met een niet zo'n goede opbouw van de schakeling (lees: de massaverbinding, zie ook fig. 3-7), maar eerlijk gezegd heb ik er geen verklaring voor. *(De probe heeft naast een weerstand, ook nog eens een capaciteit. Door de grote serieweerstand van de gebruikte brug componenten, wordt de capaciteit van de scoop probe merkbaar geladen en ontladen. Dat veroorzaakt het patroon wat zichtbaar is. Je zou eens 1k weerstanden toe moeten passen in plaats van 1M. Dan zal je zien dat het weer een blokgolf wordt -red.)*

3. Het verdraaien van de loper van de potmeter bij gelijkspanning zorgde ervoor dat het spanningsverschil tussen de 2 takken verliep van +2.3V naar -2.3V. Uitgaande hiervan was mijn verwachting dat de blokgolf zich zou omkeren als ik de loper van de ene uiterste stand naar de andere uiterste stand zou verdraaien. Dit gebeurde echter niet. Het plaatje van de blokgolf bleef hetzelfde, maakte niet uit in welke stand de loper stond. Pas na uittekenen kon ik zelf het verklaren: Ook bij blokgolf wordt een spanningsverschil gemeten. Als de loper in de ene uiterste stand staat (zeg maar de volledige weerstandswaarde) dan geldt het midden tussen R1 en R2 als nulpunt en als de loper de uiterste kant opgedraaid wordt dan wordt dat het nulpunt. Ergo: voor het plaatje blijft dat hetzelfde. (ik hoop dat ik duidelijk kan uitleggen wat ik bedoel). *(Het omdraaien gebeurt wel degelijk. Maar omdat de triggering op kanaal 1 ingesteld staat, triggert de scoop op de positieve flank. En dus zie je de blokgolf niet omdraaien, omdat de scoop na het veranderen van de fase gewoon weer op de opgaande flank triggert. Maar als je de scoop extern zou triggeren op de blokgolf - kan dat niet met de connector die ik rechts onder zie zitten? - dan zou je zien dat de blokgolf wel degelijk omdraait bij het door de nul draaien -red).*

Uit nieuwsgierigheid toch de laatste schakeling nog eens opgebouwd, maar nu met een echte wisselspanning. Ik beweerde wel dat de plaatjes ook voor een sinus/wisselspanning zouden gelden, maar ja, is dat ook zo? Gewoon, simpelweg mijn voeding opengeschroefd en vanaf de secundaire wikkeling van de trafo de wisselspanning afgetakt. Scoop erbij en eerst eens gekeken hoe het plaatje op de trafo er uit zag. Spannend is het niet, want het is eigenlijk

bekend, zie fig. 3-11. Vervolgens de trafo

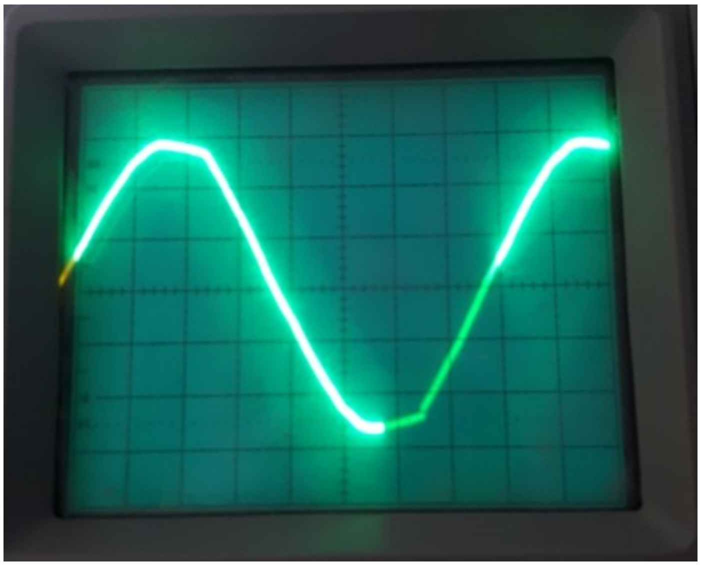


Fig. 3-11

aangesloten op de schakeling van fig. 3-4 en de potmeter in de ene uiterste stand gezet. Uiteraard de scoop aangesloten tussen de loper van de potmeter en het midden van de tak met 2 weerstanden. Dit leverde fig. 3-12 op. Al

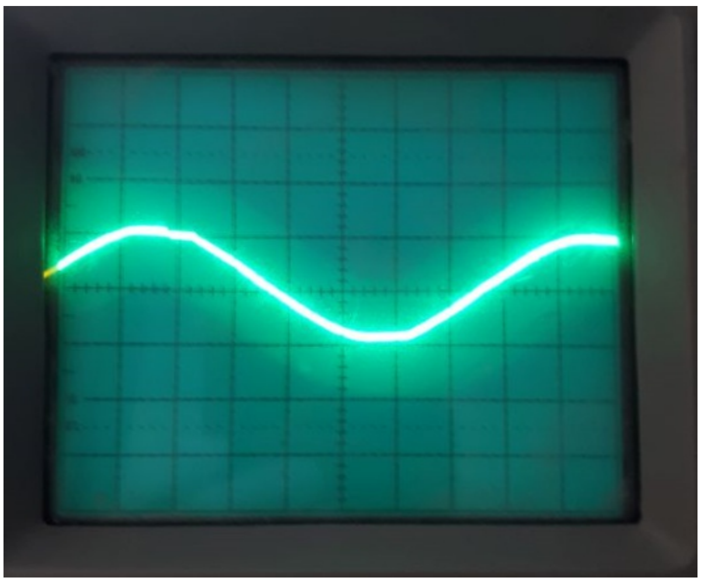


Fig. 3-12

draaiende aan de potmeter werd de amplitude tot nagenoeg een rechte streep op de scoop. Zie fig. 3-13. De potmeter doordraaien naar de andere kant verhoogde de amplitude van de sinus weer, net zoals bij de blokgolf. Dit gaf het beeld van fig. 3-14. Bij deze wisselspanning gebeurde dus hetzelfde als bij de blokgolf, behalve de amplitude. Die werd wel ongeveer

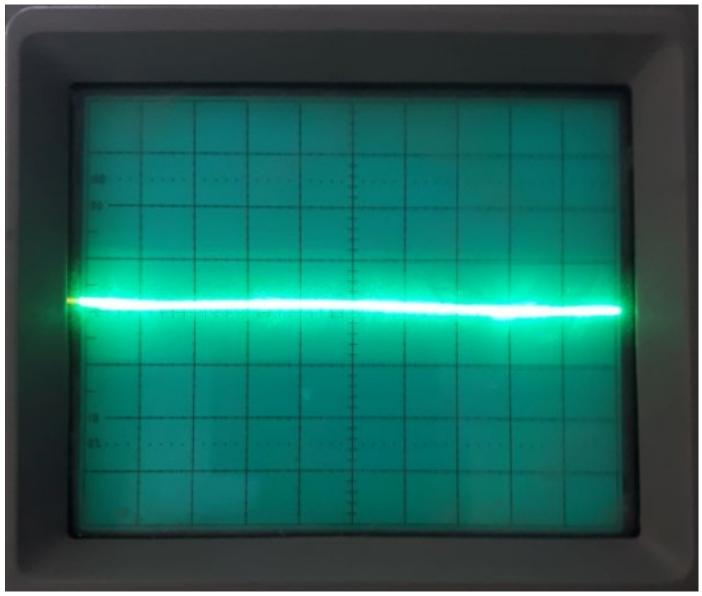


Fig. 3-13

de helft kleiner, in tegenstelling tot bij de blokgolf. Dit was ook wel wat ik verwachtte. Langs de andere kant heb ik de theorie van wisselspanning/stroom nog niet opnieuw doorgenomen, dus wellicht dat ik dan zelf ook

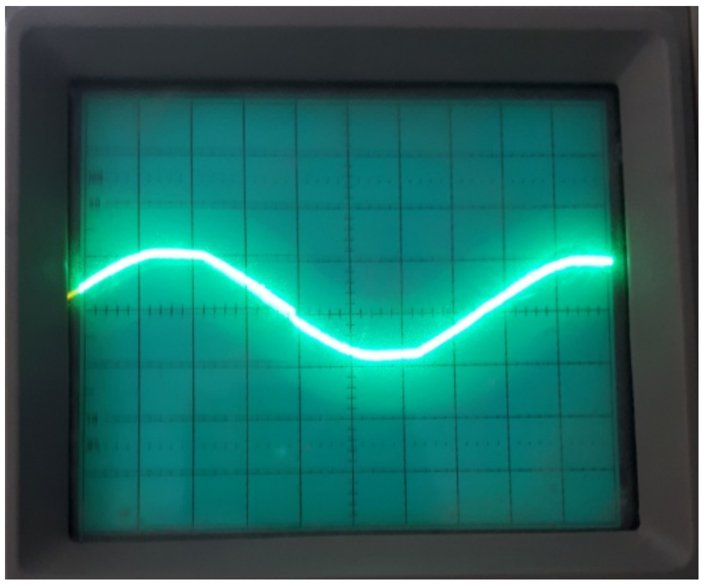


Fig. 3-14

meer kan verklaren.

Toch grappig wat zulke kleine schakelingetjes al kunnen opleveren. Volgende hoofdstuk van "Electronica, echt niet moeilijk" gaat over diodes. Ik ben benieuwd.

Reacties zijn welkom op pa5fjm@veron.nl

De VK3YE Micro 40 DSB Transceiver

Met modificaties van Dave Richards AA7EE

Voor zelfbouw zijn DSB (Dubbelzijband) transceivers eigenlijk wel zo geschikt. Er zijn geen "lastige" kristalfilters nodig, en door eenvoudige directe conversie is demodulatie ook vrij eenvoudig. Het slechte nieuws is natuurlijk dat je twee keer zoveel ruimte als normaal inneemt op de band, en vooral op 40m kan dat in drukke tijden nog wel eens een probleem zijn. Maar als je door de week als het contestgeweld verstomd is, met DSB uitkomt, kan je prima verbindingen maken.

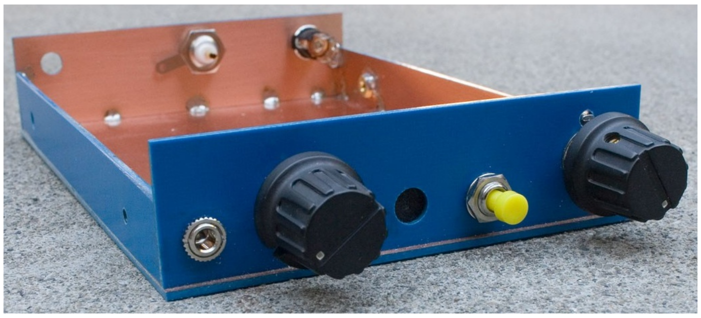
Dave AA7EE had al eens een tweetal ontwerpen geprobeerd, maar dat ging steeds goed tot en met de driver. Werd de eindtrap toegevoegd, dan stond de boel te oscilleren. Uitendelijk werd hij een keer gewezen op de VK3YE Micro 40 die door een vriend van hem gebouwd was. Hij was eerst sceptisch, en maakte wat grappen over het favoriete IC van menig zelfbouwer – de LM386. Uiteindelijk komt het altijd neer op het plaatsen van een 10uF condensator tussen de pennen 1 en 8, wat een heleboel versterking geeft maar ook een heleboel ruis en storing. Tot hij werd gewezen op een configuratie met de LM386 laagfrequent versterker die wel genoeg versterking geeft om een luidspreker aan te kunnen sturen, maar met veel minder ruis en storing dan de gebruikelijke configuraties met hoge versterking. Het project waar Dave mee bezig was werd even stil gezet, en zijn oog viel op een kastje dat hij eigenlijk gemaakt had voor de tweede batch NT7S CC-serie transceivers. Die batch werd uiteindelijk geleverd met een veel kleinere print en een kleinere behuizing, waardoor dat kastje al een jaar of twee op de plank lag, in afwachting van iets om erin te bouwen.



Enthousiast geworden over het idee van een kleine DSB transceiver in dit kastje, zocht hij de bedieningsorganen en connectors bij elkaar voor het geval hij zou besluiten om de Micro 40 te bouwen. Want daar was hij nog niet zo zeker van... Let op de kleine electret condensator miccrofoon in het midden. Het idee was dat dat de set makkelijker te gebruiken zou maken, zeker in het veld. En ondanks dat het kastje al twee jaar oud is, heeft de laklaag die hij aangebracht had het koper in prima conditie gehouden.

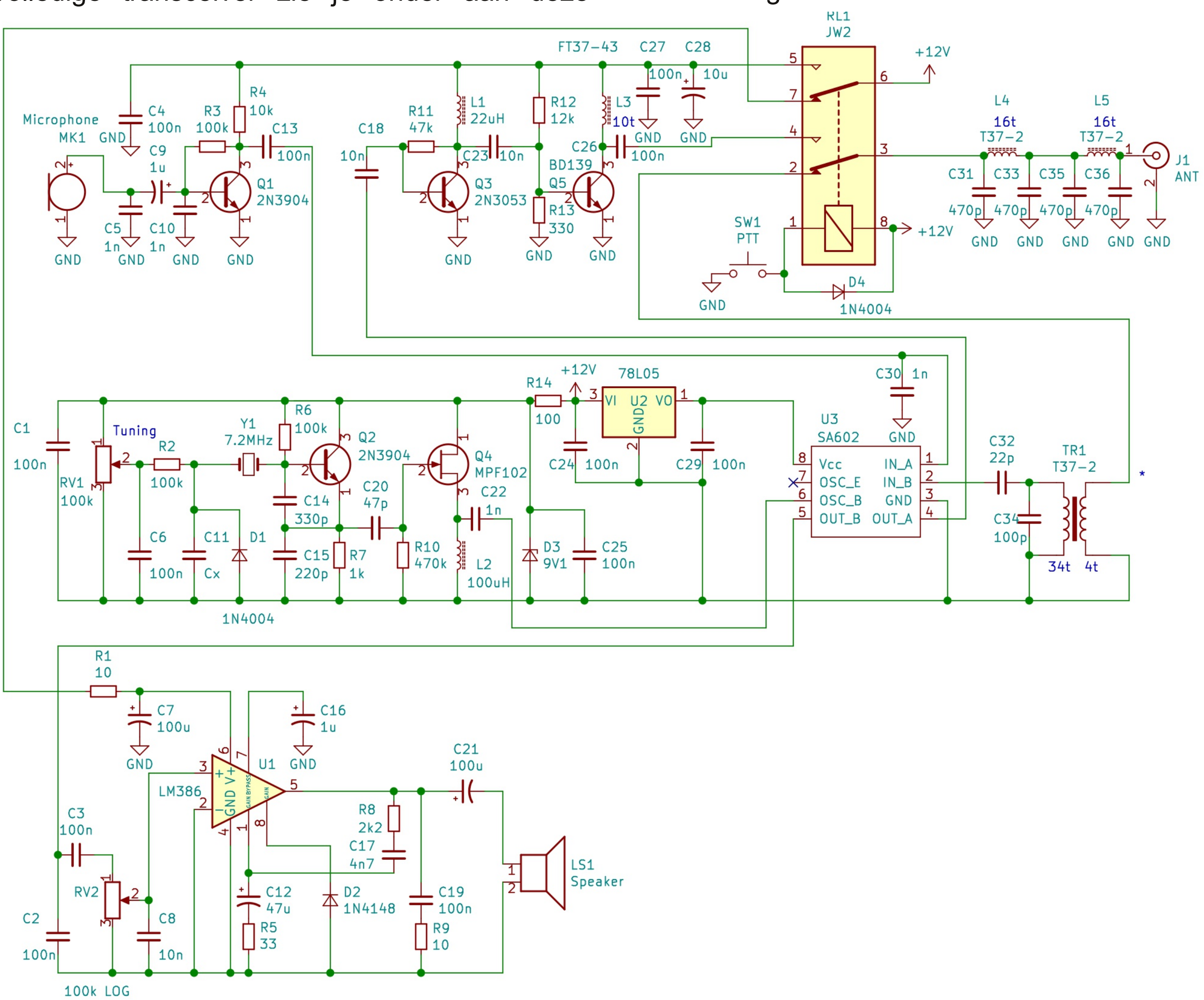


Het probleem is natuurlijk om bij het aanzien van zo'n mooi glimmend kastje met alle knoppen en connectors er al op, níét enthousiast te worden om aan de slag te gaan met de bouw van de transceiver. Let op het kleine gaatje dat boven de rechter potmeter gebouwd is om het positioneringslipje de ruimte te geven. Hij pleegde die dingen altijd af te breken (doe ik ook trouwens -red) tot hij een incidentje had met zijn Fort Tuthill 80, waar de volume potmeter los kwam en rond ging draaien. Wat er precies mis ging weet hij niet, maar een van de potmeteraansluitingen kwam ergens tegenaan,



waardoor een blauwe LED die als spanningsstabilisator gebruikt werd, opgeblazen werd. Sindsdien gebruikt Dave die positioneringslipjes wél om de potmeters op hun plaats te houden.

En vanaf dat moment had hij zich voorgenomen om zijn eerste DSB transceiver van scratch af aan op te bouwen. Hij bracht wat wijzigingen aan aan het originele schema van VK3YE en die worden hier beschreven. Het schema van de volledige transceiver zie je onder aan deze bladzijde. Eén van de problemen zat 'm in C28, meteen links van het relais. Bij het loslaten van de PTT ontlaadt deze condensator zich over de microfoonversterker en het gevolg was een luide (rondzing)piep uit de luidspreker. C28 was 220uF, maar is nu 10uF, wat het probleem zo goed als oploste. Daarnaast is C13, de koppelcondensator van de microfoonversterker naar de SA602 mixer verkleind van 1uF naar 100nF. Het originele geluid was een beetje bassig, en dat is nu ook beter. Nog een opmerking over het kristal: Dave gebruikte een Ceramic Resonator van 7.200MHz, want in Amerika loopt de band door tot 7.300MHz. Daar hebben we hier natuurlijk niets aan. Het afstembereik wat hij haalde met alleen de 1N4004 diode was 7207-7335kHz, Een bereik van meer dan 100kHz. Hamshop.cz heeft een 7.030 kristal, en door daarmee nog een zelfinductie van b.v. 10uH in



serie te zetten zou je toch wel tot de 7100 moeten kunnen komen. Dan heb je het meest gebruikte deel van de 40m band te pakken. Of: verwijder RV1, C6, R2, C11, D1, C14 en het kristal, en injecteer een digitaal VFO met een condensatortje op de basis van Q2, bijvoorbeeld een Arduino VFO met Si5351...

De diode van pin 8 van de LM386 naar massa is bedoeld om de rondzing effecten tijdens het omschakelen van zenden naar ontvangen en omgekeerd een beetje de kop in te drukken. Noodzakelijk is die niet. Maar laten we eens door de werking van de transceiver heen lopen.

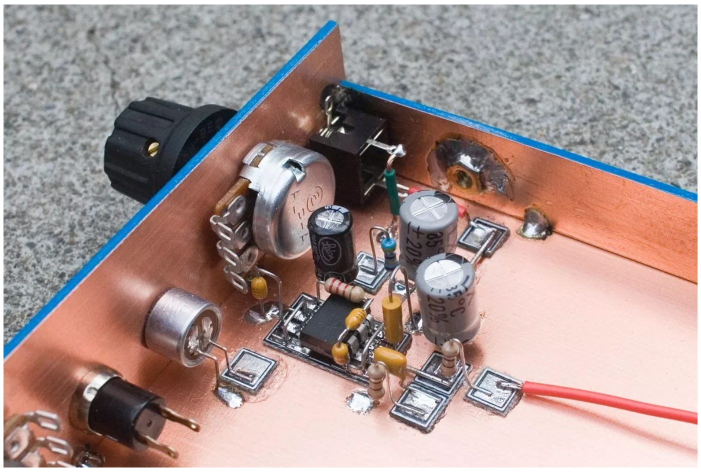
Werking van de transceiver

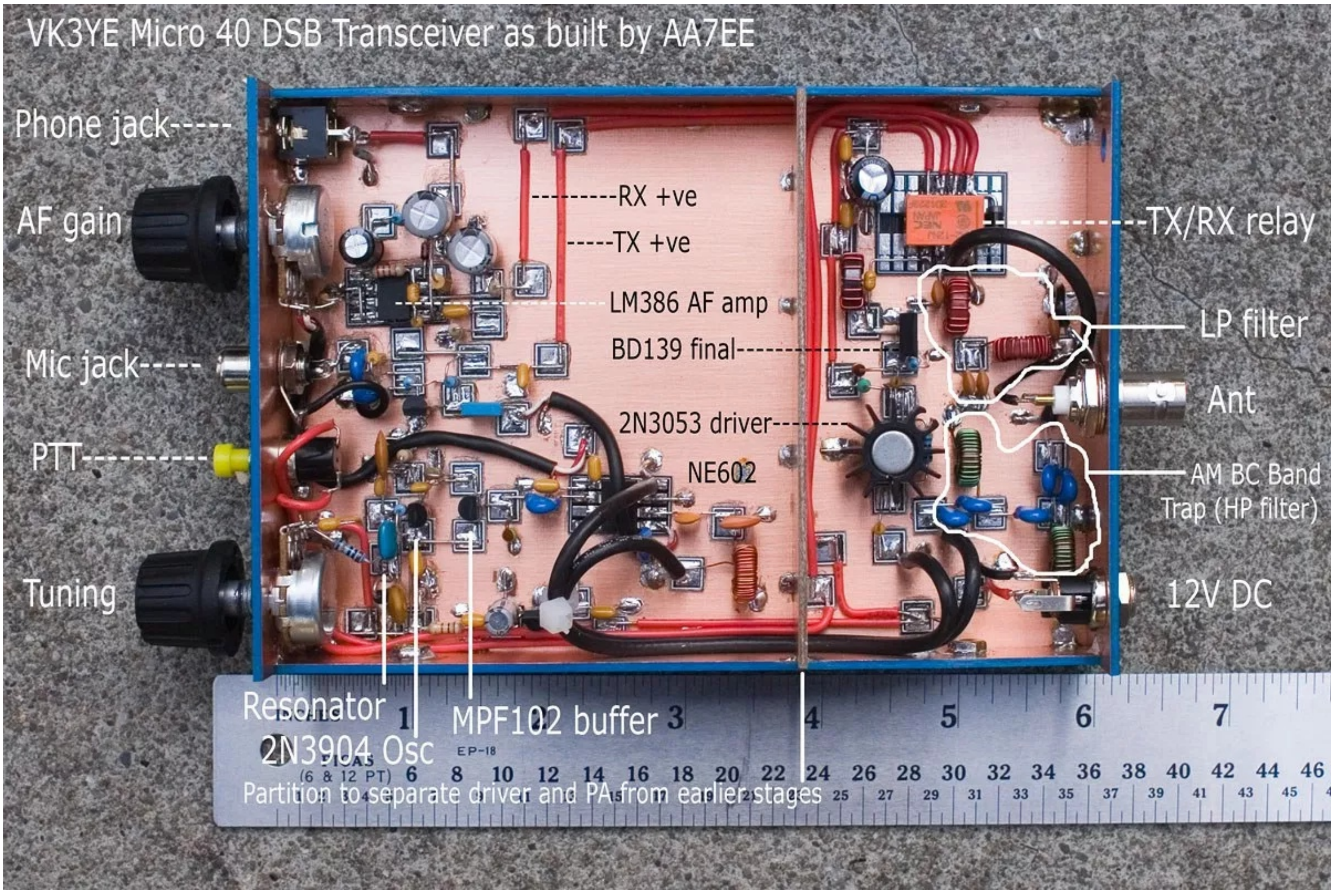
De VFO bestaande uit Q2 en buffer Q4 staan permanent onder spanning, zowel bij zenden als ontvangen. Het VFO signaal wordt aangeboden aan pin 6 van de SA602 mixer. Het antennesignaal gaat door het laagdoorlaatfilter en wordt via het rustcontact van een van de relaiscontacten toegevoerd aan TR1. De wikkelverhouding daarvan is 34:4 ofwel een factor 8,5 en daarmee wordt de antenne-spanning dan ook opgetransformeerd. De ingangsimpedantie van de SA602 is ongeveer 1,5kΩ en dat betekent dat de antennekant van de transformator ongeveer 20Ω is. Maar een kniesoor die daar op let. Het opgetransformeerde antennesignaal wordt dan toegevoerd aan pin 2 van de mixer, en het resultaat is een mengsignaal op pin 5. Dat gaat naar de volumeregelaar en de LM386 zorgt vervolgens voor voldoende vermogen aan de luidspreker. De bijzondere schakeling van de LM386 zorgt ervoor dat deze niet zo staat te ruisen zoals bij deze dingen vaak het geval is.

Bij zenden schakelen zowel de voedingsspanning als de antenne om: nu wordt de microfoonversterker van voeding voorzien. Het door Q1 versterkte microfoonsignaal wordt nu toegevoerd aan pin 1 van de mixer. Deze krijgt nog steeds VFO signaal aangeboden, en het dubbelzijband mengsignaal wordt nu afgenomen van pin 4. Dat wordt op niveau gebracht door een 2N3053 en vervolgens aangeboden aan eindtransistor Q5: een gewone BD139. Die werden in de 27Mc tijd al gebruikt om eindtrappen te repareren en daar kreeg je wel een Watt of 4 uit. Dat komt hier overigens niet uit: Dave heeft het over 800mW. Waar hij wel ongeveer 1200km mee overbrugde. Hij zegt ook dat zijn BD139 niet gekoeld hoeft te worden. Met een beetje prutsen en eventueel een extra driver tor moet een paar Watt wel haalbaar zijn, en dan moet die BD139 echt wel gekoeld worden.

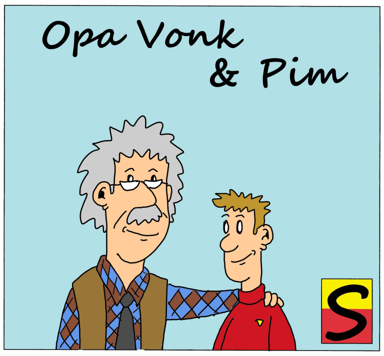
De opbouw

Zoals je hebt gezien, is het kastje geheel van printplaat gemaakt. Dat is prima spul om kastjes mee te maken, ware het niet dat een beetje printplaat tegenwoordig een vermogen kost. Vroeger haalde je op de bekende beurzen bij de eveneens bekende één-armige verkoper een pakket dubbelzijdige printplaten voor €5, maar tegenwoordig rekent hij dat per stuk. En dan ben je goedkoper uit door een fraaie behuizing uit te zoeken bij Ali. Maar laten we even uitgaan van printplaat. Wat Dave gebruikte voor het maken van verbindingen die niet aan massa liggen, zijn MeSquares en MePads. Die kan je kopen bij QRPme.com, maar tijdens het BitX project maakten we ze gewoon zelf door kleine vierkantjes te knippen van een plaat dubbelzijdig printplaat, en die met secondenlijm op de plekken te lijmen waar we dingen aan elkaar moesten knopen. Werkt net zo goed.

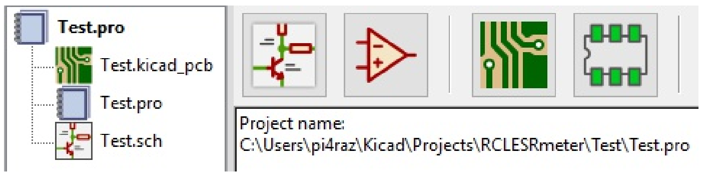




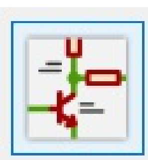
Hierboven zie je de volledige opbouw van de transceiver, inclusief beschrijving wat waar zit. Aan de lineaal kan je zien dat het een best wel kleine transceiver is. Natuurlijk moet het op de band wel een beetje rustig zijn wil je QSO's kunnen maken: om te beginnen is het vermogen absoluut QRP (800mw DSB, en dat is te vergelijken met 400mW SSB. Overigens maakte Gert PE0MGB tijdens de bouw van zijn transceiver ook al verbindingen op de stuurtrap met 200mW, dus het kan makkelijk) maar belangrijker is dat een ontvanger volgens het Directe Conversie principe geen onderscheid kan maken tussen zijbanden, en dat de selectiviteit bepaald wordt door het laagfrequent. Dus moet je eigenlijk een scherp 2700Hz filter in het audio hebben en dat zit hier niet in. Je hoort op enig moment 6kHz tegelijk, en dat kunnen in een drukke band best twee stations zijn. Anyway, niet in het weekend dus met die eeuwige contesten, maar op een doordeweekse dag is het prima verbindingen maken met dit soort setjes. Zie dit als een goede basis voor verdere experimenten: een resonator die voor onze banden geschikt is, zal lastig te vinden zijn. Maar je kunt aan de gang met een VXO met een kristal ergens tussen 7030 en 7150, of met een digitale VFO met Arduino zoals ik eerder al aangaf. Daarmee kan je meteen de hele band bestrijken. Verder kan je zelf een laagfrequent filter toevoegen om de selectiviteit te verbeteren. Je kunt een extra driver trap toevoegen om het vermogen wat op te peppen; een BD139 kan best wel beter dan 800mW. Verder kan je dan experimenteren met het toevoegen van een AGC schakeling, waar je meteen de spanning voor een S-meter van af kunt leiden. Denk er wel om dat als je een Electret microfoon gebruikt, de microfooningang met een weerstandje aan de voeding gehangen moet worden en C9 moet worden omgedraaid. Een electret heeft spanning nodig om te kunnen werken. Overigens heeft Dave de interne microfoon uiteindelijk toch vervangen door een chassisdeel voor het aansluiten van een externe microfoon: de interne microfoon gaf problemen met "motorboaten" en een externe microfoon loste dat op. Succes met experimenteren!



Pim stond over Opa's schouder mee te kijken terwijl Opa bezig was om een schema te tekenen. Hij keek het enige tijd aan, en zei toen: "Wat handig! Ik vroeg me al af hoe u altijd aan die mooie schema's kwam, maar zo doet u dat dus. Hoe heet dat programma?" Opa zei zonder op te kijken van zijn scherm: "KiCad. Er zijn meer programma's waar je schema's mee kunt tekenen. Vroeger waren die programma's onbetaalbaar, maar tegenwoordig zijn die als freeware op internet te krijgen. TinyCad is een hele bekende. Die heeft Opa ook heel lang gebruikt, maar het voordeel van KiCad is dat je vanuit het maken van je schema meteen door kunt naar het ontwerpen van een gedrukte bedrading, ofwel een print. En vroeger moest je dan allemaal Gerber files maken voordat je een print kon laten maken, maar zelfs dat hoeft niet meer". Pim keek met grote ogen naar Opa's beeldscherm. "Kan ik dat ook leren?" vroeg hij. "Natuurlijk", zei Opa. "En voor jou als Windows-hater zal het een opluchtig zijn om te weten dat KiCad er niet alleen voor Windows is, maar ook voor Linux en MacOS. Kijk maar eens op de site http://kicad-pcb.org/, daar vind je de downloads voor het operating systeem naar keuze. Je hebt wel een beetje ruimte nodig op je computer, want de Windows download is b.v. al 1,1GB. Maar dan heb je ook wat." Pim rende Opa's piephok uit, dat die naam verdiend had door de morseseinen die met enige regelmaat het hele huis vulden, om een kwartier later weer binnen te vallen met een laptop in zijn hand: "En nu? Hoe teken ik schema's?" riep hij hijgend van het rennen. "Rustig", begon Opa. "Start het programma eerst eens op. Je zult zien dat je het in het Nederlands kunt draaien, zodat ook als je het Engels niet goed beheerst, je het programma makkelijk kunt gebruiken. Het eerste wat je nu doet is een Project aanmaken. Dat kan door in Bestand te kiezen voor Nieuw en dan Project, maar met Control-N gaat dat in één keer. Dat zal je vaker zien met het programma: er zijn veel sneltoetsen waarmee je de gewenste functie meteen kunt aanroepen, zonder door allemaal drop-down menu's te moeten klikken. Geef je project nu een naam, en sla dat op als KiCad project bestand. Dat doet hij trouwens automatisch als je het project een naam geeft. Vanaf nu is dat de basis voor alle handelingen die je gaat verrichten. Laten we zeggen dat we het project even "test" noemen. Als je dat hebt ingevuld als naam en hebt opgeslagen, zie je dat hij automatisch een soort boomstructuur aanmaakt met de projectonderdelen die je straks nodig hebt:

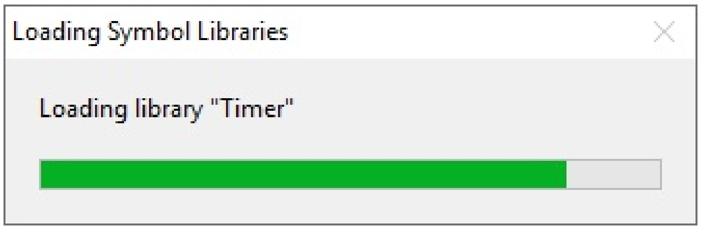


Test.pro is je project bestand. Test.sch is je schema bestand, waar straks je schema in opgeslagen wordt. En Test.kicad_pcb is het bestand waar straks je gedrukte schakeling in opgeslagen wordt. Handig hè? Uiteraard beginnen we eerst met het tekenen van een schema. Klik daarvoor op het Schema symbool boven in de balk. Er opent nu een nieuw venster met een kader waarin je je schema kunt tekenen.

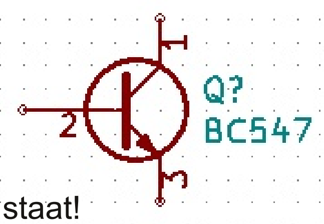


Met je muiswiel kun je inzoomen. Als het al niet aan staat, kun je bij weergave "Show Grid" aanvinken. Dat helpt je om onderdelen uit te lijnen zodat ze mooi tegenover elkaar komen te liggen. Laten we eens een audio versterkertje gaan tekenen. Ik gebruik een transistor als voorversterker en een LM386 IC als eindversterker. Het goede nieuws is dat KiCad al geleverd wordt met een uitgebreide bibliotheek aan onderdelen. Vandaar dat het ook zoveel ruimte inneemt. En zit een onderdeel niet in een van de bibliotheken, dan kan je het onderdeel zelf maken. Dat doen we later nog wel een keertje. Ik ga eerst de transistor op zijn plaats zetten, en daarna ga ik de componenten daar omheen toevoegen. Ik stel voor dat jij op je laptop hetzelfde doet: teken maar met me mee, dan krijg je vanzelf gevoel bij hoe het programma werkt.

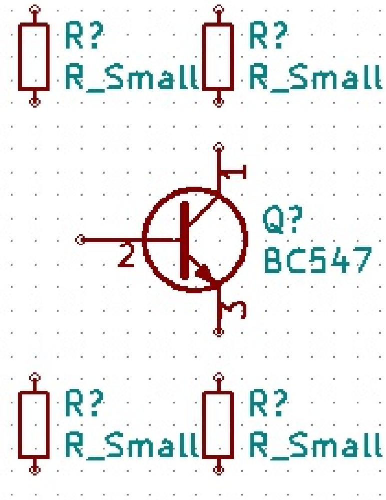
We gaan beginnen met een transistor te plaatsen. Dat doe je door bovenin op "Plaatsen" te klikken, en dan kies je "Symbol". Maar je kunt ook de sneltoets "A" gebruiken. De eerste keer dat je dat doet, zal het progamma de beschikbare bibliotheken laden:



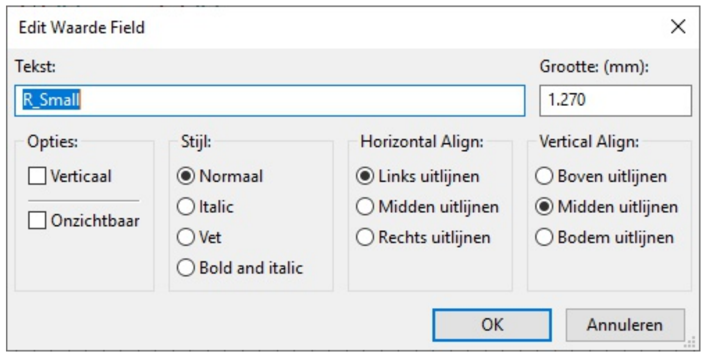
Na het laden opent er een venster waarin je het gewenste symbool kunt zoeken. Je kunt op een algemene term zoeken, bijvoorbeeld op NPN. Dan kan je een generieke transistor kiezen. Maar het programma kent al een heleboel standaard transistoren. Kijk maar: als we zoeken op BC547, dan staat deze er gewoon in. Het voordeel van een transistor die al bekend is, is dat het programma straks ook weet welke "footprint" erbij hoort. De footprint van een onderdeel is eigenlijk zijn vorm zoals hij straks op de print komt. Dat bespreken we wel als we een print gaan ontwerpen. Heb je het onderdeel gekozen, dan kan je deze met de muis over je tekening bewegen. Op de plek waar je het onderdeel wil hebben, klik je met je muistoets en dan staat hij op zijn plek. Je eerste onderdeel staat!



Vervolgens gaan we weerstandjes toevoegen in de emitter en collector, en een spanningsdeler op de basis om de transistor in te stellen. Je klikt weer op "Plaats" en "Symbool" (of de letter a) en nu zoek je naar R. Het programma vindt nu meteen een weerstand voor je. Maar Opa gebruikt altijd kleinere weerstanden, zodat er wat meer op een tekening past. Dus kies ik nu R_Small. Merk op dat de gebruikte afkortingen wél Engels zijn: de R staat natuurlijk voor Resistor, het Engelse woord voor Weerstand. Plaats deze nu op hun plek: als je na het plaatsen op Enter drukt, komt weer het menu naar voren met je laatste keuze en zo kan je snel meerdere weerstanden plaatsen. Je moet nu iets hebben zoals op het volgende plaatje:

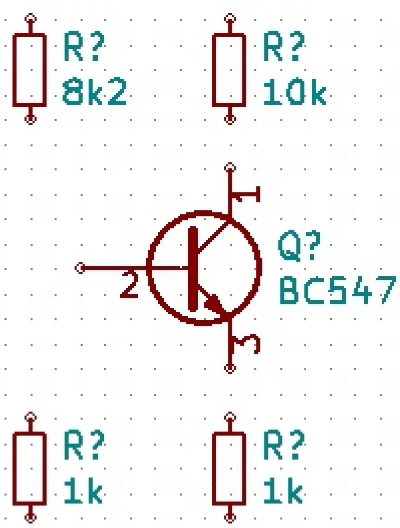


Zoals je ziet hebben de weerstanden nog geen waarde, en ook het volgnummer is nog niet ingevuld. De waarden gaan we nu invullen, maar de volgnummers laten we nu voor wat het is. Ik zal je later wel tonen waarom. Druk op "Escape" zodat je het vizier weer terugkrijgt als muispointer. Hou nu de muis boven de weerstand linksboven op de tekening en druk op de toets "V" (van Value, ofwel de waarde). Er opent nu een scherm waarin je de waarde van de component kunt aanpassen.

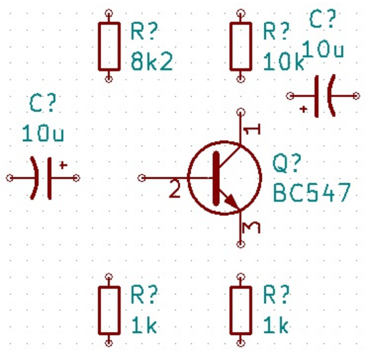


Verander nu R_Small in 8k2. Maak van de weerstand rechtsboven 10k, en de twee onderste weerstanden zijn beiden 1k. Daarmee heb je de waarden van je onderdelen benoemd. Nu moeten we nog koppelcondensatoren toevoegen voor de in- en de uitgang van deze voorversterkertrap.

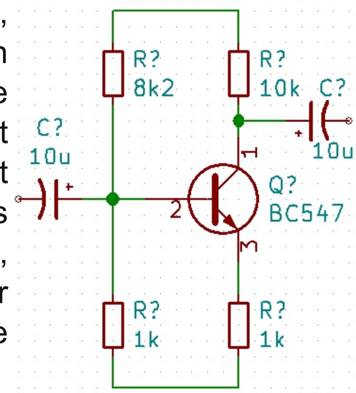
Druk nu weer op de toets "a" en zoek naar "C". Je krijgt nu een rijtje condensatoren te zien, en we kiezen voor CP1_Small. CP staat voor Capacitor Polarized en dat is een elco. Voor een gewone condensator zou ik C_Small



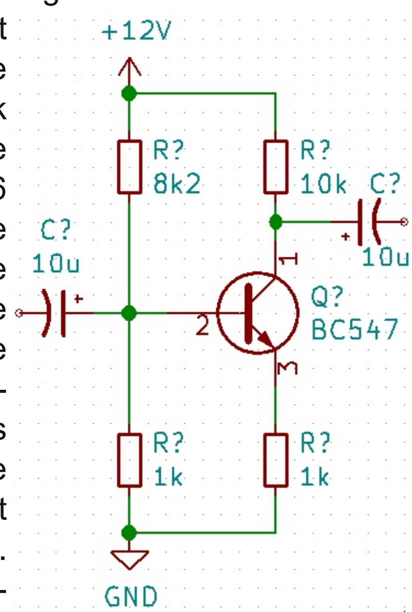
gekozen hebben. Opa houdt van kleine onderdelen: die kan je makkelijk dicht op elkaar zetten. Maar zoals je ziet, kan je de condensatoren alleen vertikaal zetten. Druk op "Escape" tot je je vizier weer terug hebt, hou deze boven de condensator en druk op R van Rotate (draaien). Elke keer dat je op R drukt, draait het component 90 graden en zo kan je deze de gewenste richting geven. Maar er is een grote kans dat de onderdelen niet staan waar je ze hebben wilt. Hou weer de muis boven de desbetreffende component en druk op M van Move (verplaatsen). Nu kan je de onderdelen op hun plek zetten. Geef de condensatoren met V de waarde 10u en dan moet je ongeveer dit plaatje hebben. Merk op dat de waarde van de 10u condensator rechts boven best wel in de weg zit bij de 10k weerstand. Als je je muis boven de waarde 10u houdt en weer de letter M indrukt, dan kan je de waarde verplaatsen naar een plek waar hij niet in de weg zit. Datzelfde geldt voor de onderdelenreferentie: met de M kan je alles verplaatsen.



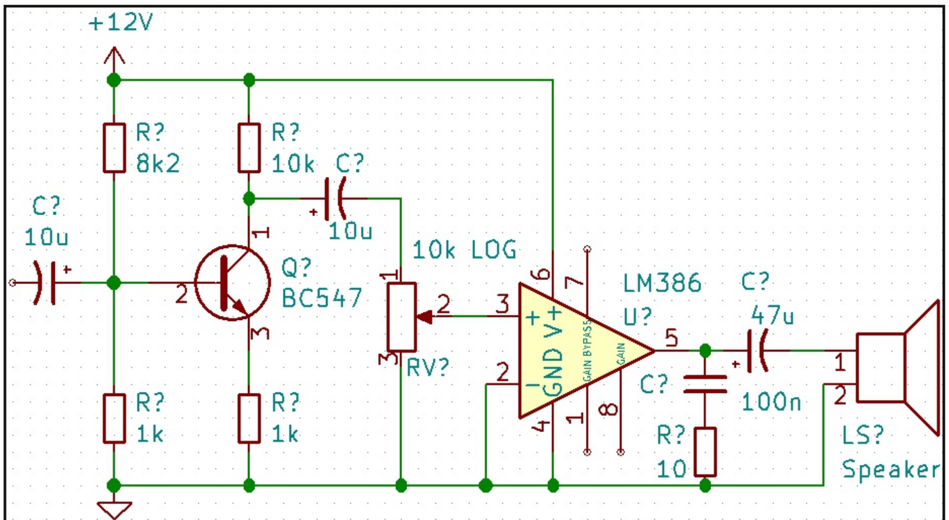
Tot nu toe staan er alleen maar losse onderdelen op je werkblad. Het wordt tijd om de onderdelen met elkaar te verbinden. Houd daarvoor de muis boven een van de aansluitingen van een onderdeel, bijvoorbeeld de ingangscondensator van 10u. Druk op "W" en je muiscursor verandert in een pen. Nu kan je naar rechts naar de basis van de transistor, en klik op het rondje. De twee onderdelen zijn nu verbonden, en de muiscursor blijft een pen. Door nu op de weerstand van 8k2 te klikken kan je die verbinden met de lijn tussen de condensator en de basis van de transistor. Daar ontstaat nu een bolletje ten teken dat het hier geen kruisende lijnen betreft, maar een verbinding. Klik weer op het bolletje en verbindt de drie onderdelen met de weerstand van 1k. Als je een hoek om moet, kan je met de muis op de plek klikken waar je af wil buigen, en dan loopt de lijn niet meer met je cursor mee maar gaat verder waar je geklikt heb. Doe dat met het verbinden van de bovenste en onderste twee weerstanden. Gaat het fout, druk dan op Esc en de pen wordt weer een vizier. Je moet dan weer opnieuw "W" indrukken om te beginnen met een nieuwe draad. Wil je een lijn weghalen, hou dan je muiscursor boven die lijn en druk op "Del". De lijn zal dan verdwijnen. Weet het programma niet wat je wil deleten, bijvoorbeeld op een kruising, dan krijg je een popup waarin het programma vraagt wat je bedoelt. Heb je alles aan elkaar verbonden, dan moet je ongeveer het volgende plaatje hebben.



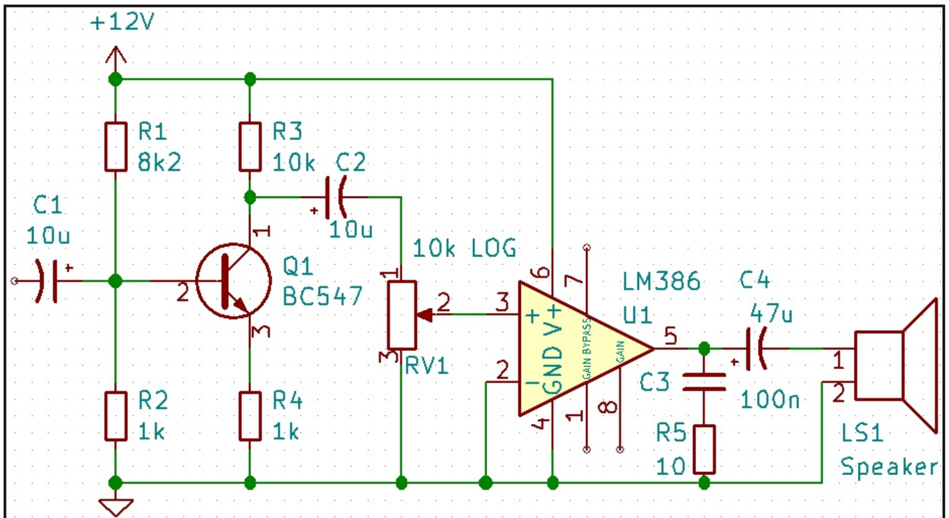
Nu moeten we de schakeling nog van voeding voorzien. Want de onderdelen zitten wel aan elkaar, maar er staat nog nergens spanning op. Voeding voeg je toe met de letter P van Power. In de seach box die dan opent, type je GND. Nu krijg je het massa symbool. Klik op OK en plaats de massa op de onderste verbindingslijn. Merk op dat de cursor weer een pen blijft. Klik nu op de bovenste lijn en de search box verschijnt weer, met nu GND als voorkeuze. Maar nu typen we +12V en klikken op OK. Plaats de voeding op de bovenste lijn en nu is de schakeling van voeding voorzien. Nu kunnen we verder met het opbouwen van de schakeling. Zoek naar 386 en je vindt de LM386 versterker. Voor de potmeter neem je R_POT en voor de luidspreker zoek je op SPEAKER. Zo-als je ziet zit alles er al in en kan je de onderdelen met elkaar verbinden. Verplaats waar no-



dig met de letter M de referenties, wijzig de waarden met V en verplaats die eventueel ook met de M en uiteindelijk heb je de versterker helemaal klaar en moet het ongeveer hierop lijken.

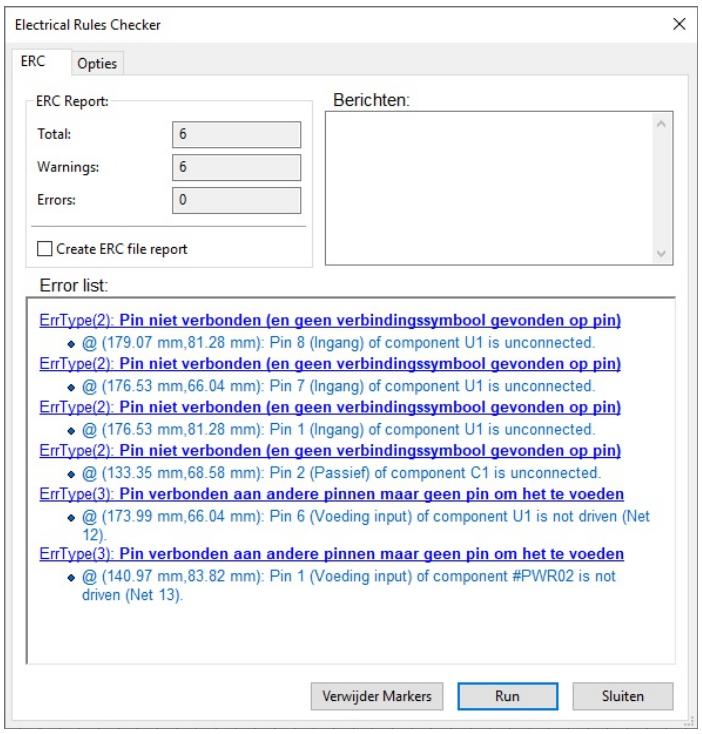


Het schema voor de annotatie



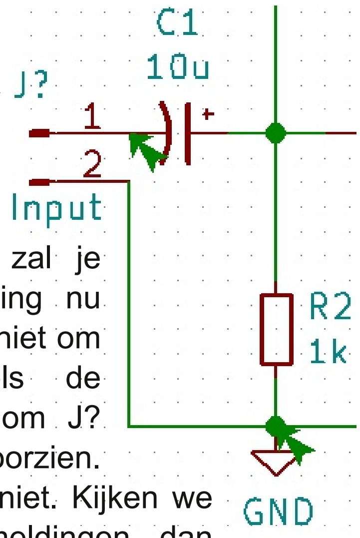
Het schema na de annotatie

De volgende stap is om alle onderdelenreferenties van een nummer te voorzien. Dat hoef je gelukkig niet met de hand te doen. Kies daarvoor in de menubalk "Tools" en klik op "Annotate schematic". Je hoeft aan de voorkeursinstellingen niets te wijzigen: klik nu op Annotate en zoals je zult zien worden alle onderdelen nu netjes van een volgnummer voorzien. Hoe makkelijk wil je het hebben. En was het dat? Nee, nog niet... Nu moeten we nog controleren of we geen fouten hebben gemaakt. Ook dat kan het programma voor je controleren. Daarvoor voeren we een Electrical Rules Check  uit door op het rode kevertje in de menubalk te klikken. Nu opent het Electrical Rules Checker (ERC) venster. Druk nu op Run en... Oh jee! een hele lijst met fouten, zie hier rechts. Wat je nu ook in je schema ziet verschijnen, is groene pijlen. Die geven aan waar de problemen zitten. Sommige fouten liggen voor de hand: Hoewel het voor ons mensen duidelijk is dat condensator C1 de ingang van de schakeling is, ziet de ERC alleen maar dat er een component in de lucht hangt. En als je er straks een print van wil maken, zal er toch ook ergens een gaatje in de print moeten zitten waar je het draadje van de ingang aan kunt solderen. Dus daar moet een connector aan. Overigens: als je op een foutmelding klikt, centreert het schema op de desbetreffende fout. Je ziet dus meteen waar de fout zit. Aan de ingang plaatsen we nu een connector. Type weer A zodat de bibliotheek opent, en zoek op Conn. Je ziet dan een reeks

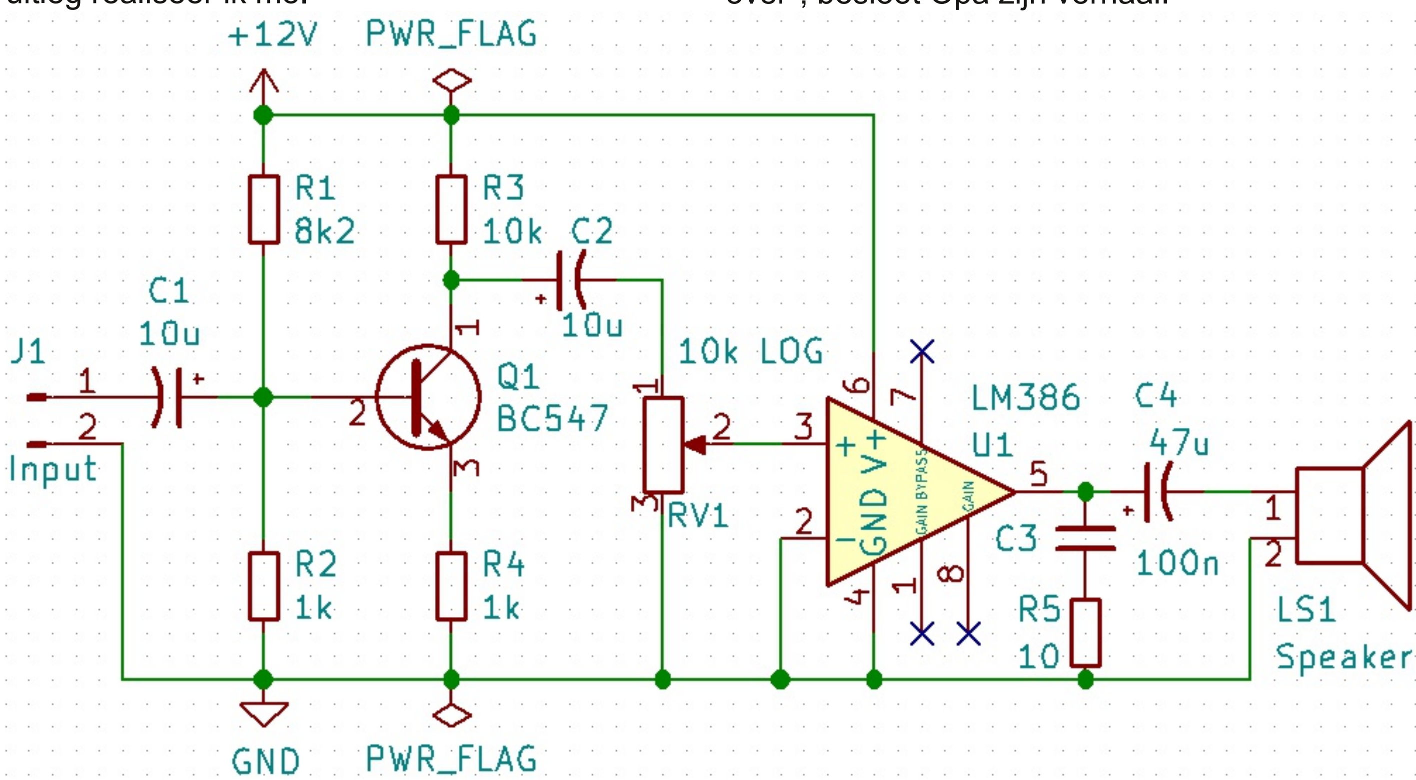


De foutmeldingen na het draaien van de ERC

connectoren verschijnen die je kunt gebruiken. Kies voor Conn_01x02_Male en plaats die op de ingang. Als je nu de ERC opnieuw draait, zal je zien dat die foutmelding nu verdwenen is. Vergeet niet om eerst weer in Tools de annotatie te draaien, om J? van een nummer te voorzien. Anders werkt de ERC niet. Kijken we naar de overige foutmeldingen, dan zijn er drie die melden dat er aansluitingen van IC U1 niet aangesloten zijn. Dat klopt: de pennen 1, 7 en 8. Die moeten ook niet aangesloten zijn, maar de ERC weet niet of dat zo bedoeld is, of dat jij dat vergeten bent. Niet gebruikte aansluitingen moeten in KiCad als zodanig benoemd worden. Gebruik daarvoor de Plaats No-Connect vlag, wat je in één keer kunt doen met de Q op het toetsenbord. Je ziet dan een blauw kruisje aan de aansluiting en daarmee is de aansluiting gemarkeerd als niet-gebruikt. En dan blijven er twee foutmeldingen over: 'Pin verbonden aan andere pinnen maar geen pin om het te voeden'. Dat behoeft enige uitleg realiseer ik me.



Bij het definiëren van een component moet je aangeven of de aansluitingen van die component een uitgang, ingang of bidirectioneel zijn. De ERC gebruikt dat om te zien of je nergens uitgangen aan elkaar knoopt wat voor problemen kan zorgen. Voedingsaansluitingen zijn allemaal ingangen, maar er is nergens een punt dat die ingangen ook van voeding voorziet. Daar is een truc voor: Druk op P voor het plaatsen van een Power symbool, en zoek op General. Of PWR. Je ziet dan Power Flag General. Die plaats je op de +12V lijn en op de GND lijn, en die "voeden" dan de voedingsaansluitingen. Als je nu de ERC draait, zal je zien dat alle fouten zijn opgelost. Je schakeling is nu gereed voor de voorbereidingen van het maken van een print", besloot Opa. "Het duizelt me wel een beetje", zei Pim. "Maar ik denk dat als je dit onder de knie hebt, het best leuk is om mijn eigen schema's te kunnen tekenen". Opa knikte. "Mooie schema's zijn ook handig als je later terug wil kijken hoe je iets gemaakt hebt. Maar het grootste voordeel vind ik dat ik meteen door kan om een print te ontwerpen. En vroeger moest je dan aan de gang met belichtingskasten en etsmiddelen, maar tegenwoordig stuur je je KiCad bestand op en drie weken later heb je je print. Maar daar vertel ik je de volgende keer over", besloot Opa zijn verhaal.



APRS iGate met Arduino

Zoals in de vorige RAZzies al beschreven, hadden we het idee om een eenvoudige, goedkope iGate te maken, gebaseerd op een Arduino. Dit omdat er in een aantal gebieden in Nederland best wel wat witte vlekken zijn (vooral als je naar het noorden rijdt), en het goed voor te stellen is dat amateurs niet permanent een computer en een set (of op zijn minst een ontvanger) kwijt willen zijn voor het doorgeven van andermans data. Onze expedities naar Liechtenstein zijn altijd uitgelezen momenten om ideeën om te zetten naar de praktijk, en dus werd de uitdaging aldaar onderzocht op haalbaarheid. Het eerste wat uit het onderzoek kwam, was dat de Arduino het niet in zijn eentje voor elkaar ging krijgen om naar het audio uit de DRA818 ontvangstchip te luisteren, daar data van te maken, die coderen zodat aprs.fi het snapt, een Wifi verbinding op te zetten, contact te maken en te houden met aprs.fi en de ontvangen data daar afleveren. Zo'n krachtige processor heeft een Arduino nou ook weer niet.

Na enige literatuur geraadpleegd te hebben, bleek er al software te bestaan die van een Arduino een TNC kan maken, ofwel een Terminal Node Controller. Een TNC wordt gebruikt om mee te doen in AX.25 netwerken, en, zoals je wellicht weet, is APRS geheel gebaseerd op het AX.25 protocol. Het enige wat we hoefden te doen, was de TNC software in de Arduino laden, het - enigszins versterkte - audio van de DRA818 ontvanger chip aan de Arduino toevoeren en op een lijntje van de Arduino komt de ruwe data gewoon naar buiten. Maar die moest vervolgens nog op een of andere manier aan aprs.fi aangeboden worden.

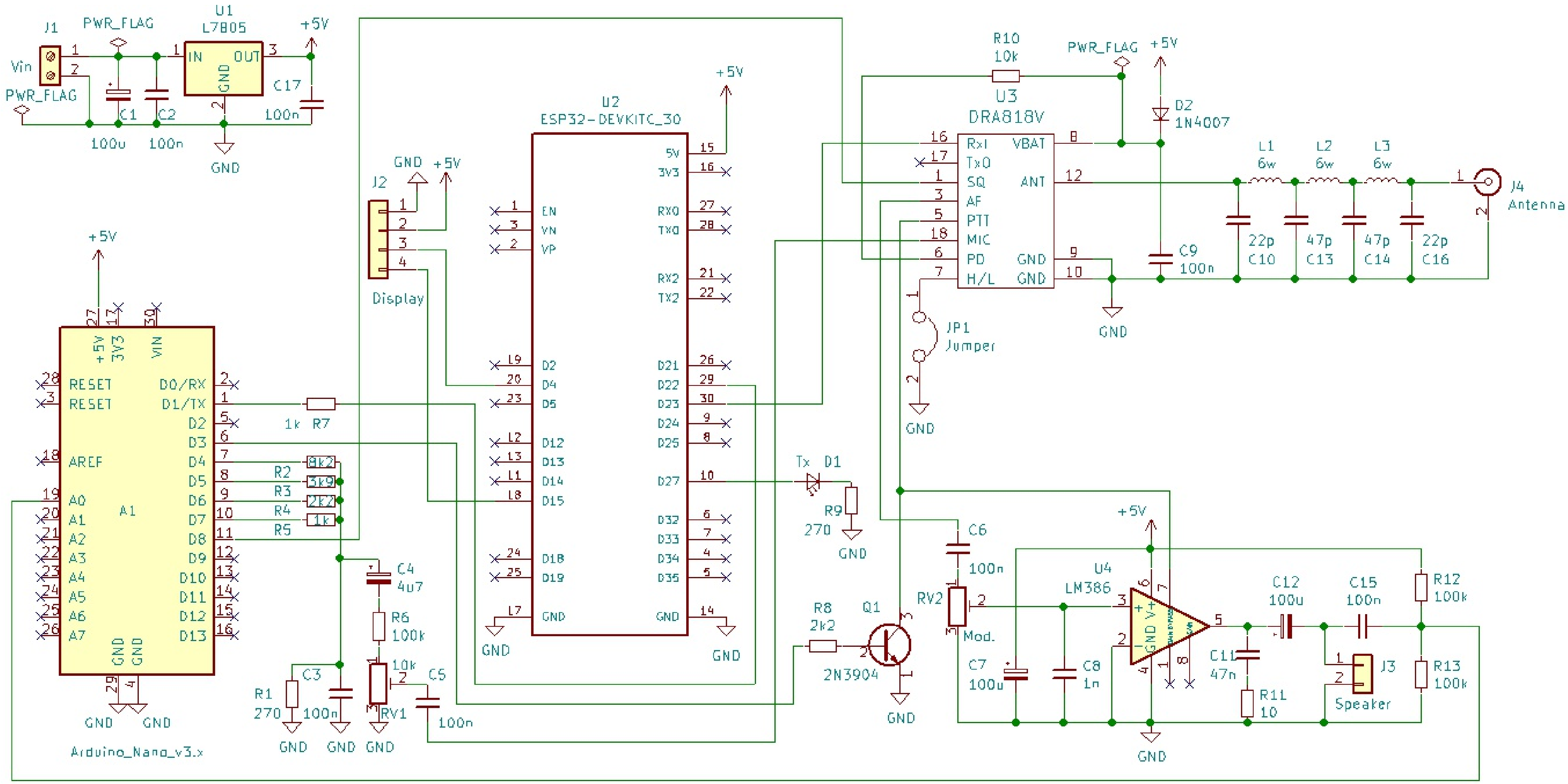
Nou had Robert PA2RDK al eerder gespeeld met ESP32 processoren. Die dingen hebben al WiFi en Bluetooth aan boord, en dat scheelt gewoon een hoop gepruts met losse WiFi modules, zoals we dat bij de onweerdetectors nog deden. Voor de prijs hoef je het niet te laten:

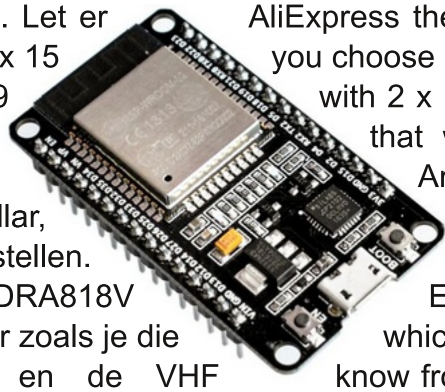
APRS iGate with Arduino

As described in the previous issue of the RAZzies, we had the idea to create a simple, inexpensive iGate based on an Arduino processor. This is because there are quite a number of white spots in certain areas in the Netherlands (especially if you are driving north), and it is imaginable that amateurs do not want a computer and a transceiver (or at least a receiver) permanently dedicated to APRS, just for relaying other people's data. Our expeditions to Liechtenstein are always perfect moments to turn ideas into practice, and so we used our time in Liechtenstein to see whether the idea was feasibel. The first thing that came out of the assessment was that the Arduino on its own was not going to be able to listen to the audio from the DRA818 receiver chip, convert the audio into data, encode the data into a format that is going to be understood by aprs.fi, setup a Wifi connection, establish and maintain a connection with aprs.fi and deliver the received data there. The processor of the Arduino is just not powerful enough for all those tasks.

After consulting some literature, it turned out that there was already software available on the internet that could turn an Arduino into a TNC, short for Terminal Node Controller. A TNC is used by amateurs to participate in AX.25 networks, and, as you may know, APRS is entirely based on the AX.25 protocol. All we had to do was load the TNC software into the Arduino, apply the - somewhat amplified - audio from the DRA818 receiver chip to the Arduino and the raw data would just come out of one of the Arduino's I/O lines. But that data still had to be presented to the aprs.fi website in one way or another.

Robert PA2RDK had been playing before with ESP32 processors. Those things already have WiFi and Bluetooth on board, and that just saves a lot of messing around with separate WiFi modules, like we did with the lightning detectors. You don't have to get a mortgage to buy one: at


J1
Vin
PWR_FLAG
C1 100u
C2 100n
U1 L7805
IN OUT GND
C17 100n
+5V
GND
U2 ESP32-DEVKITC_30
J2 Display
U3 DRA818V
RxI TxO SQ AF PTT MIC PD H/L VBAT ANT GND
R10 10k
D2 1N4007
C9 100n
L1 6w
L2 6w
L3 6w
C10 22p
C13 47p
C14 47p
C16 22p
J4 Antenna
JP1 Jumper
Tx D1
R9 270
R8 2k2
Q1 2N3904
C6 100n
RV2 Mod.
C7 100u
C8 1n
U4 LM386
C11 47n
R11 10
C12 100u
C15 100n
J3 Speaker
R12 100k
R13 100k
A1 Arduino_Nano_v3.x
R7 1k
R2 8k2
R3 3k9
R4 2k2
R5 1k
C4 4u7
R6 100k
RV1 10k
C5 100n
C3 100n
R1 270



bij Ali kosten ze ongeveer €3,50. Let er wel op dat je er een neemt met 2x 15 pootjes: er zijn er ook met 2x 19 pootjes en die gaan niet op de print passen. De Arduino's kosten bij Ali nog geen twee dollar, dus die kan je gelijk mee bestellen. Daarnaast moet je nog een DRA818V hebben: dat is de hele transceiver zoals je die kent van de APRS tracker en de VHF transceiver die we eerder publiceerden. En hoe dat dan allemaal aan elkaar zit, zie je op het schema op de vorige bladzijde.

Laten we beginnen bij de antenne: daar komen de APRS packets immers binnen. Eerst kom je dan een laagdoorlaatfilter tegen. Dat is voor de ontvangst niet noodzakelijk, maar de schakeling is voorbereid om als volledig functionele digipeater te werken en dan moet hij ook kunnen zenden, vandaar het laagdoorlaatfilter. Het signaal komt dan bij de DRA818V transceiver chip. De laagfrequent uitgang daarvan gaat via instelpotmeter RV2 naar een laagfrequentversterker met een LM386. Dat had ook een Op-amp kunnen zijn, maar deze was op voorraad in Liechtenstein... Het voordeel is dat er ook een luidspreker op aangesloten kan worden, zodat je kunt horen of er packets binnenkomen. In een drukke omgeving hou je dat niet heel lang vol...

De uitgang van de LM386 gaat naar analoge ingang A0 van de Arduino. Zoals eerder gezegd, hebben we die geladen met een reeds bestaande HEX file die van de Arduino een KISS TNC maakt. (Meestal vertaald als Keep It Simple, Stupid). Hierbij wordt de binnenkomende data direct doorgegeven aan de RS232. Maar zo direct bleek dit niet te zijn: Berichten zitten ingeklemd tussen 2 FEND karakters (0xC0) en worden zo ook afgeleverd aan de RS232, Dat heeft best wel wat tijd gekost omdat het bericht dus niet met een CR/LF maar met een FEND word afgesloten. Volgens Robert is dat geen correcte implementatie van KISS in de door ons gebruikte HEX file, maar we moesten het er mee doen.

AliExpress they cost around € 3.50. Make sure you choose one with 2x 15 pins: they also exist with 2 x 19 pins that will not fit on the PCB that we designed for the circuit. The Arduinos cost less than two dollars at AliExpress, so you can order them right away along with the ESP32. You also need a DRA818V: which is the entire transceiver as you know from the APRS tracker and the VHF transceiver that we published earlier in our magazine. And you can see how that all works together on the diagram on the previous page.

Let's start at the antenna connector: after all, that's where the APRS packets arrive. First you come across a low-pass filter. This is not necessary for receiving, but the circuit is prepared to work as a fully functional digipeater and then it must also be able to transmit, hence the low-pass filter. The signal then arrives at the DRA818V transceiver chip. The low-frequency output of the transceiver chip goes via trimmer potentiometer RV2 to a low-frequency amplifier consisting of the well-known LM386. That could also have been an Op-amp, but the LM386 was in stock in Liechtenstein... The advantage is that it can also be connected to a speaker, so you can hear if packets are coming in. In a busy APRS environment you probably won't use this feature for long...

The output of the LM386 goes to analog input A0 of the Arduino. As mentioned earlier, we programmed the Arduino with an already existing HEX file that turns the Arduino into a KISS TNC. (Usually translated as Keep It Simple, Stupid). Hereby the incoming data is passed directly to the Arduino's RS232 output. But it turned out not to be so directly at all: Messages are sandwiched between 2 FEND characters (0xC0) and are delivered this way to the RS232, which took quite some time because the message is not terminated with a CR/LF but with a FEND character. According to Robert PA2RDK, that is not a correct implementation of KISS in the HEX file we use, but we had to deal with it.

Er waren verschillende implementaties van een TNC te vinden, maar allemaal alleen maar beschikbaar in HEX. Dit stukje zelf bouwen leek een hele klus, dus vandaar de gesplitste oplossing met een Arduino waar de HEX file in geladen wordt en de ESP32 voor de verwerking van de data. Daarbij heb je dan ook het voordeel dat e.e.a. ook als KISS TNC te gebruiken is.
Zie ook: http://www.ax25.net/kiss.aspx.
De TNC biedt de RS232 data vervolgens aan aan de ESP32: van D1 van de Arduino naar D22 van de ESP32. In de ESP32 moet de ruwe APRS data omgezet worden naar bruikbare informatie. In een loop in de software wordt de data in een array geplaatst totdat er een FEND karakter voorbij komt. Deze wordt dan vervangen door een LF (LineFeed) en dan wordt de informatie in de buffer verwerkt. In het eerste stuk van de data, tot het karakter 0x03 voorbij komt, zijn alle bytes 1 bit naar links geschoven. Dit wordt weer naar rechts geschoven om de oorspronkelijke data te herstellen. Het bericht wordt tenslotte geconverteerd naar het door de APRS-IS site gewenste fixed length formaat. Hierbij blijft de inhoud van de velden zoals comment en destination onveranderd, het maakt dus niet uit of er Mic-E of standaard berichten worden verwerkt. Je kunt namelijk op verschillende manieren berichten verzenden in APRS. Omdat de standaard tekstformaat berichten als relatief lang beschouwd werden, is op ingenieuze wijze gebruik gemaakt van het feit dat APRS berichten als Unnumbered Information frames worden verzonden in AX.25, waarbij het "aan" adresveld misbruikt kan worden om b.v. positie-informatie in op te slaan, waardoor het packet korter wordt. Dat heet Mic-Encoder, kortweg Mic-E genoemd. De iGate hoeft dat dus niet te decoderen, maar alleen maar door te geven aan de APRS-IS site. De ESP32 verbindt met je WiFi en maakt via de WiFi verbinding met de APRS-IS site, waar je positie informatie aan kunt bieden. Hierbij moet worden ingelogd met een call en een APRS PASSCODE. De passcode kan hier worden gegenereerd:
https://apps.magicbug.co.uk/passcode/

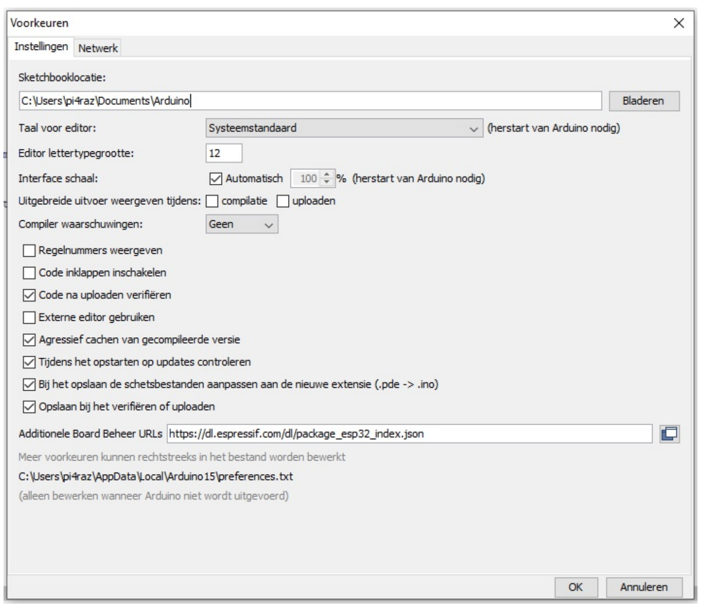
There were several implementations of a TNC on the internet, but all only available in HEX format. Building this part ourselves seemed quite a job, hence the split solution with an Arduino which will be programmed with the HEX file and the ESP32 processor for processing the data. Thereby you also have the additional advantage that you can also use the Arduino as a KISS TNC.
See also: http://www.ax25.net/kiss.aspx.
The TNC subsequently presents the RS232 data to the ESP32: from D1 of the Arduino to D22 of the ESP32. In the ESP32, the raw APRS data must be converted into useful information. In a loop in the software, the data is placed in an array until a FEND character is found in the data. This character is then replaced by an LF (LineFeed) and then the information is processed in the buffer. In the first part of the data, until the character 0x03 is encountered, all bytes are shifted 1 bit to the left. These bytes are shifted 1 bit to the right again to restore the original data. The message is finally converted to the fixed length format desired by the APRS-IS site. The content of the fields such as comment and destination remains unchanged, so it does not matter whether Mic-E or standard messages are processed. There are different ways to send messages in the APRS protocol. Because the standard text format messages were considered to be relatively long which extends the transmit time, ingenious use was made of the fact that APRS messages are sent as Unnumbered Information frames in AX.25, whereby the "to" address field can be misused to store position information for example, making the packet shorter. This is called Mic-Encoder, or Mic-E for short. The iGate does not have to decode the Mic-E format, but simply pass it on to the APRS-IS site. The ESP32 connects to your WiFi and connects via the WiFi to the APRS-IS site, where you can present position information. At the APRS-IS site, you have to log in with a callsign and an APRS PASS CODE. The pass code can be generated here:

https://apps.magicbug.co.uk/passcode/

Via de USB van de Arduino Nano kan de TNC ook los worden gebruikt. Inkomende data is hier beschikbaar en aangeboden data wordt verzonden mits deze conform spec is, dus met een FEND karakter aan begin en eind. Daarvoor is de hardware voorbereid: D3 van de Arduino stuurt de PTT van de DRA818V aan, en D4 t/m D7 genereren in combinatie met de weerstanden R1 t/m R5 en condensator C3 als een soort primitieve DA-converter het audio voor de DRA. Na het aanbieden van een bericht aan de Arduino schakelt deze automatisch de zender in om het bericht te versturen. Je hoeft dus verder niets te doen.

Installatie van het ESP32 package

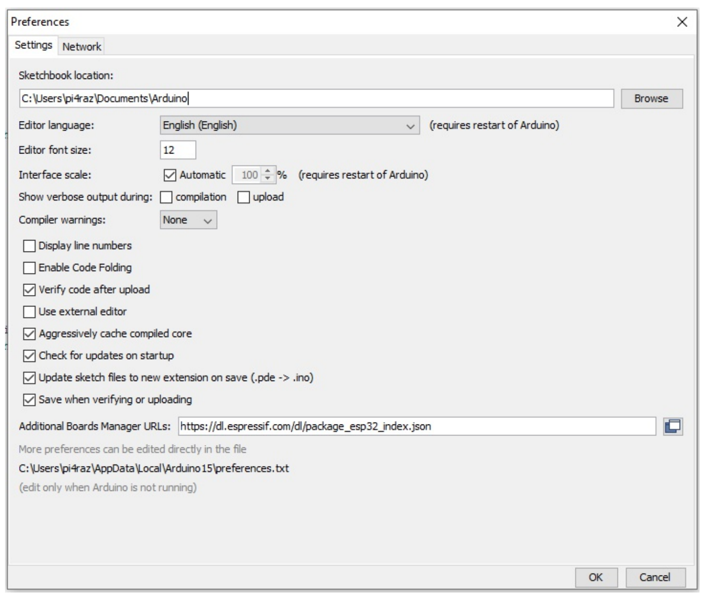
De ESP32 is geen Arduino. Het goede nieuws is dat je de Arduino ontwikkelomgeving wel kunt gebruiken om de ESP32 code te compileren, mits deze minimaal versie 1.8 heeft. Daarvoor moet er een additioneel pakket geïnstalleerd worden dat je kunt vinden op Github via deze link: http://bit.ly/2EmowGw
Hier vind je de instructies voor het installeren van de ESP32 add-on. Ik heb de Nederlandse versie, en die gaat als volgt: Open eerst je Arduino omgeving. Klik links boven op Bestand en dan Voorkeuren. In het venster dat dan opent, zie je onderaan "Additionele Board Beheer URLs". Kopieer daar de link die je op Github ziet staan, als volgt:



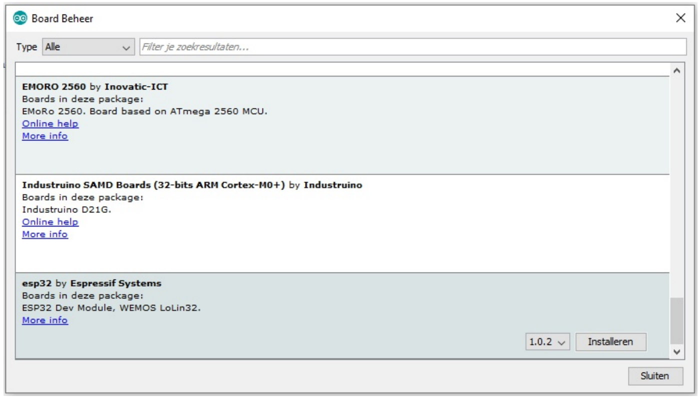
The TNC can also be used separately via the Arduino Nano's USB interface. Incoming data is available here and data offered to the TNC is sent provided it is in conformity with the specifications, so with a FEND character at the beginning and end. The hardware is prepared for this: D3 from the Arduino controls the PTT of the DRA818V, and D4 to D7 generate in combination with resistors R1 to R5 and capacitor C3 the audio signal to the DRA as a kind of primitive DA converter. After offering a message to the Arduino, it automatically switches to transmit to send the message. So you don't have to do anything additional.

Installation of the ESP32 package

The ESP32 is not an Arduino. The good news is that you can use the Arduino development environment to compile the ESP32 code, provided the IDE has at least version 1.8. For this an additional package must be installed which you can find on Github via this link:
http://bit.ly/2EmowGw
Here you will find the instructions for installing the ESP32 add-on. I have the Dutch version, but it works as follows: First open your Arduino environment. Click on File at the top left and then Preferences. In the window that opens, you will see "Additional Board Management URLs" at the bottom of the window. Copy the link that you see on Github, as follows:



(Gelukkig kan je PDF's vergroten HI). Klik op OK en hij begint ongeveer 375MB aan data binnen te halen. Als dat klaar is, staat de software erop. Maar nu nog in de IDE installeren. Klik op Hulpmiddelen, Board, en in de lijst die dan opent, staat bovenaan "Board beheer". Klik daar op en er opent nu een venster met allerlei boards. Scroll helemaal naar beneden en daar zie je nu **esp32** by **Espressif systems**. Klik op de link more info en rechtsonder verschijnt nu een knop "installeren". Klik daarop, en vervolgens wordt de esp32 software geïnstalleerd.



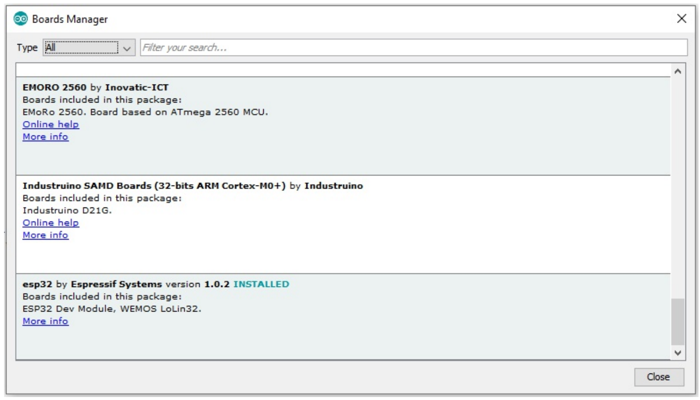
Als de installatie gereed is, vind je nu onder Hulpmiddelen - Board een hele lijst ESP32 devices. Kies de ESP32 DEV module. Nu kan je de software van de iGate compileren.

Configureren

Is dat allemaal goed verlopen, dan kan de iGate geconfigureerd worden. Voor degenen die onze APRS tracker gebouwd hebben, zal het menu bekend voorkomen. De volgende instellingen kunnen gemaakt worden:

SSID - Niet te verwarren met de APRS SSID: deze is van je WiFi access point. Maximaal 25 karakters.
pass - Het wachtwoord van je WiFi. Eveneens maximaal 25 karakters.
callSign - De roepnaam waarmee je iGate op aprs.fi verschijnt, b.v. PA3CNO-10
modemChannel - Het kanaal waarop geluisterd wordt. Dit is het aantal 12,5kHz stapjes gerekend vanaf 144.000: voor 144.800 is dat kanaal 64
oledTimeout - De tijd dat het Oled display aan blijft na een ontvangen packet (default 5)

(Fortunately you can enlarge PDFs HI). Click on OK and the IDE will start collecting approximately 375MB of data. When the download has finished, the software is on your harddisk. Now you have to install it in the IDE. Click on Tools, Board, and in the list that opens, click "Board management" on top of the list. In the window that opens, scroll all the way down and you will now see **esp32** by **Espressif systems**. Click on the 'more info' link and a "install" button will now appear at the bottom right. Click on it, and then the esp32 software will be installed.



When the installation has finished, you will now find a whole list of ESP32 devices under Tools - Board. Select the ESP32 DEV module. Now you can compile the iGate software.

Configure

If all went well, the iGate can be configured now. For those who have previously built our APRS tracker, the menu will appear familiar. The following settings can be made:

SSID - Not to be confused with the APRS SSID: this is the SSID of your WiFi access point. Maximum 25 characters.
pass - Your WiFi password. Also a maximum of 25 characters.
callSign - The nickname with which your iGate appears on aprs.fi, e.g. PA3CNO-10
modemChannel - The channel on which the iGate will be listening. This is the number of 12.5 kHz steps calculated from 144,000: for 144,800 the channel is 64
oledTimeout - The time that the Oled display stays on after a received packet (default 5)

updateInterval - de tijd in seconden waartussen de iGate zijn eigen call aanmeldt op aprs.fi
passCode - je gegenereerde APRS Key (maximaal 6 karakters)
latitude - de latitude van de positie van je iGate, maximaal 9 karakters
longitude - de longitude van de positie van je iGate, maximaal 10 karakters
PHG - het PHG getal (zie tekst). Maximaal 9 karakters
APRSIP - de URL waar de iGate zich moet aanmelden. Default rotate.aprs.net
APRSPort - Het poortnummer dat hij moet gebruiken voor het aanmelden. Default 14580
destination - De APRS destination om aprs.fi tevreden te houden. Moet met AP beginnen. Default APRAZ1, maximaal 7 karakters.

Maak de invoer niet langer dan het aantal gespecificeerde karakters: anders kan er van alles mis gaan. Heb je echt meer ruimte nodig, dan moet je in het programma de structure StoreStruct aanpassen.

De variabele PHG (staat voor Power Height Gain) beschrijft een aantal stationsvariabelen in verkort formaat, zoals het vermogen, de antenne hoogte, antenne versterking en antenne richting. De beschrijving van de parameters en een web tool om de variabele te bepalen vind je hier:
http://bit.ly/2YGaZRX
Gebruik je 10W met een antenne met 0dB gain die omnidirectioneel is en op 20 voet boven de straat staat, dan wordt je PHG code PHG3100/. Overigens is de literatuur niet duidelijk over de PHG code. Je ziet ze met 4 getallen zoals hier, en met 5, b.v. PHG01003/, waarbij het laatste getal het aantal bakens per uur is dat je iGate zelf uitzendt. Het advies is om dat op 2 te zetten.

Bouwen van de iGate

Zeker als je niet van plan bent om de iGate ooit als Digipeater te gebruiken, kan je de onderdelen die daarvoor nodig zijn, weglaten. Je houdt dan de Arduino, de ESP32, de DRA818V en de LM386 over. Dat is best op een stukje experimenteerprint te zetten, zie foto. Er zitten

updateInterval - the time in seconds between which the iGate reports its own call on aprs.fi
passCode - your generated APRS Key (maximum 6 characters)
latitude - the latitude of the position of your iGate, maximum 9 characters
longitude - the longitude of the position of your iGate, maximum 10 characters
PHG - the PHG number (see text). Maximum 9 characters
APRSIP - the URL where the iGate has to register itself. Default rotate.aprs.net
APRSPort - The port number to use for registering. Default 14580
destination - The APRS destination to keep aprs.fi happy. Must start with AP. Default APRAZ1, maximum 7 characters.

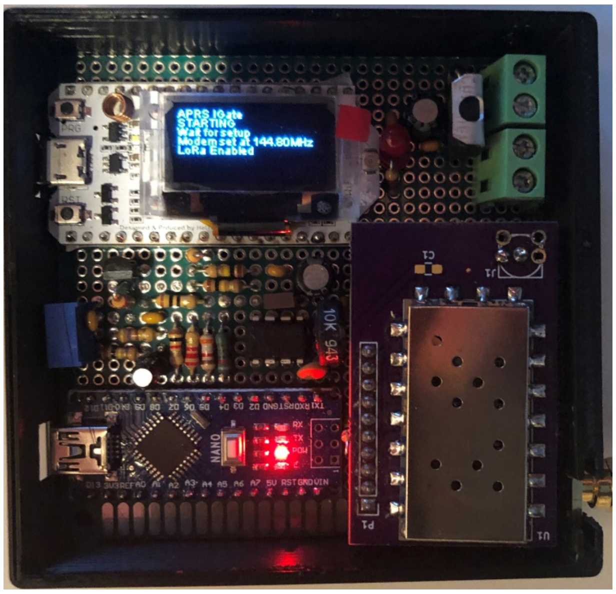
Do not make the input longer than the number of specified characters: otherwise various things may go wrong. If you really need more space, you need to adjust the StoreStruct structure in the program.

The variable PHG (stands for Power Height Gain) describes a number of station variables in a short format, such as the power, the antenna height, antenna gain and antenna direction. The description of the parameters and a web tool to determine the variable can be found here:
http://bit.ly/2YGaZRX
If you use 10W into an antenna with 0dB gain that is omnidirectional and 20 feet above the street, your PHG code becomes PHG3100/. The literature is not clear about the PHG code. You see both 4 digit numbers like this, and 5 digit numbers, e.g. PHG01003/, where the last digit is the number of beacons per hour that your iGate broadcasts. The advice is to set that to 2.

Building the iGate

Especially if you do not intend to ever use the iGate as a Digipeater, you can omit the parts that are required for that. You then have only the Arduino, the ESP32, the DRA818V and the LM386. Those are easy to mount on a piece of Vero board, see the picture. You need only a few

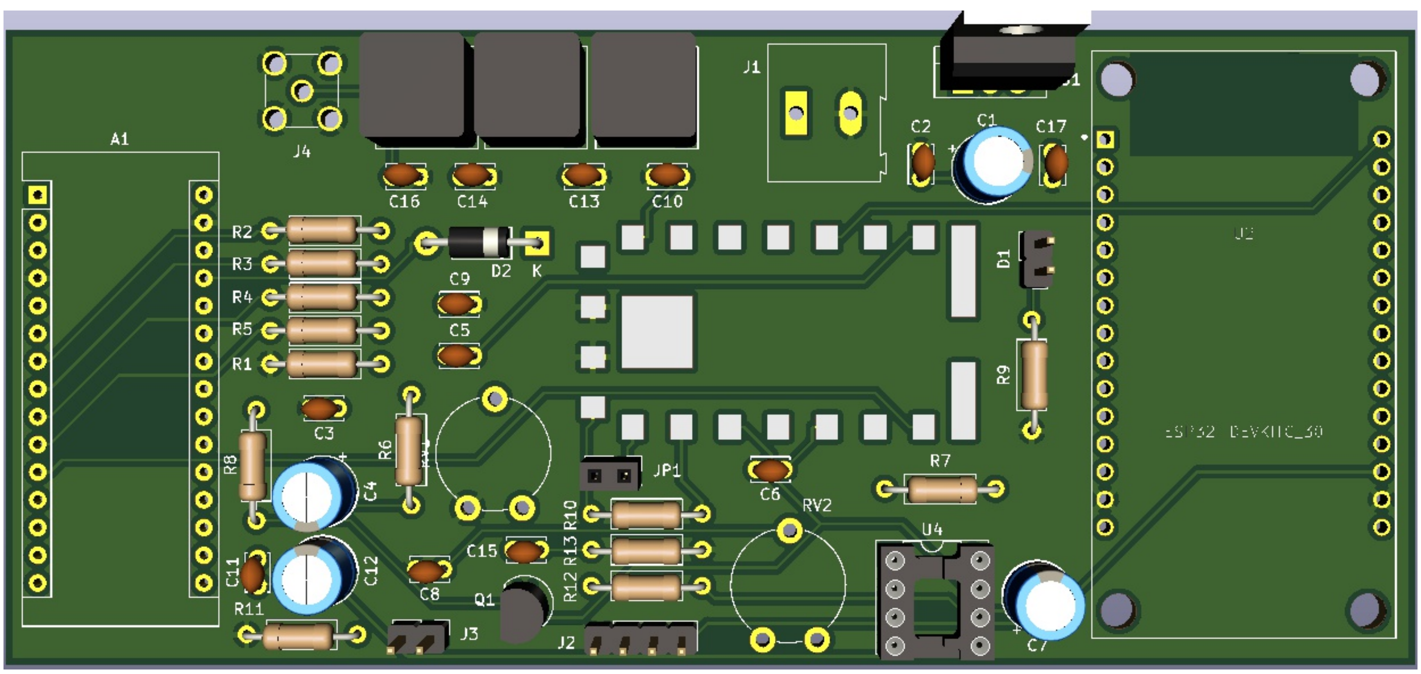
maar een paar draadjes tussen de Arduino, ESP32 en DRA818V. Alleen de LF versterker met de LM386 heeft wat componenten die op de print gesoldeerd moeten worden. Ben je niet goed in schakelingen bouwen op experimenteer-bord, dan hebben we een print ontworpen voor de iGate. Een 3D impressie zie je hieronder. We gaan deze eenmalig als kit uitbrengen: er was voldoende belang-stelling voor. Nieuws hierover volgt binnenkort.



wires between the Arduino, the ESP32 and the DRA818V. Only the LF amplifier with the LM386 has some components that need to be soldered to the board. If you are not good at building circuits on Vero board, then we have designed a printed circuit board for the iGate. You can see a 3D impression below. We are going to release this one-off as a kit: there was sufficient interest in it. News about this will follow soon.

Werkt het? Jazeker. Om het te demonstreren, hebben we een pagina gemaakt waarop je de door Robert PA2RDK ontvangen packets kunt zien: http://onweer.pi4raz.nl/igate/packets.php (de site van de onweerdetector is even misbruikt omdat er dan geen certificaten nodig zijn om de site te benaderen). Klik je op een station, dan zie je de track (of de positie) van dat station op de kaart. Onze eigen aprs.fi gebaseerd op OpenStreetMaps. Ondanks dat de antenne van Robert's iGate niet hoger staat dan een meter of 3, is het bereik heel behoorlijk. Hopelijk zullen meer amateurs de iGate gebruiken om de witte vlekken op de kaart te laten verdwijnen!

Does it work? Hell yes. To demonstrate it, we have created a page on which you can see the packets received by Robert PA2RDKs iGate: http://lightning.pi4raz.nl/igate/packets.php (the site of the lightning detector has been temporarily abused because no certificates are needed then to access the site). If you click on a station, you will see the track (or the position) of that station on the map. Our own aprs.fi based on OpenStreetMaps. Although the antenna of Robert's iGate is not higher than about 3 meters, the range is very reasonable. Hopefully more amateurs will use the iGate to make the white spots on the map disappear!

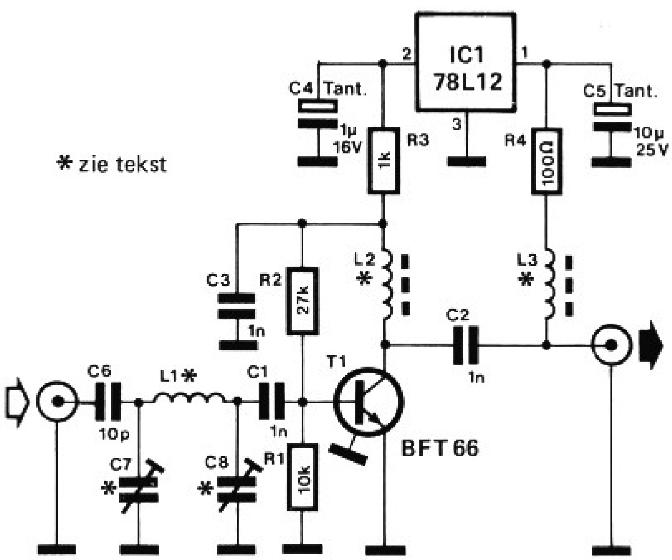


Tot slot: de software zal beschikbaar komen op Robert's Github pagina:
https://github.com/pa2rdk?tab=repositories

Finally: the software will be available on Robert's Github page:
https://github.com/pa2rdk?tab=repositories

Aanvulling 2m antenneversterker

Naar aanleiding van mijn artikel over de 2m antenneversterker kreeg ik een mail van Paul Boersma PA5PB. Ik had in het artikel met de natte duim de instelling van de transistor berekend om te zien of er niet teveel spanning op de collector zou komen. Daarbij verwaarloosde ik de basisstroom van de transistor en in veel gevallen mag dat ook. Als de spanningsdeler in de basis voldoende laagohmig is, maak je daar geen heel grote fout mee. Maar in dit geval was de spanningsdeler niet voldoende laagohmig, en dus was de afwijking van mijn berekening best groot. Hier volgt nog even hoe ik het wél had moeten berekenen:



Dit was het originele schema. De spanningsregelaar had ik weggelaten en een simpele stabilisator voor in de plaats gezet met een zener van 5,6V met twee LEDs in serie zodat ik op ongeveer 9,6V uit kwam. Dat is dus de spanning op C4 en aan de bovenzijde van R3. Voor de stroom door R3 geldt nu:

$$I_{R3} = I_c + I_b + I_{R1}$$

Alle stroom die er loopt moet immers door R3, en dat is de collectorstroom, de basisstroom, en de stroom die nog door R1 naar massa loopt. Kijken we naar wat er door R2 loopt:

$$I_{R2} = I_b + I_{R1}$$

En tot slot de stroom door R1. Op de basis van de transistor kan nooit meer dan 0,7V staan, want dat is de doorlaatspanning van de basis-emitterdiode. Dat betekent voor de stroom door R1:

$$I_{R1} = \frac{0,7V}{10k\Omega} = 70\mu A$$

Verder geldt, als we aannemen dat h_{FE} 100 is:

$$I_c = h_{FE} * I_b \Rightarrow I_c = 100 * I_b$$

De spanning over R3 is gelijk aan de waarde van R3 maal de stroom die er doorheen loopt, en daarvoor geldt:

$$\begin{aligned} U_{R3} &= R3 * (I_c + I_b + I_{R1}) \\ &= 1k\Omega * (100 * I_b + I_b + 70\mu A) \\ &= 1k\Omega * (101 * I_b + 70\mu A) \end{aligned}$$

Voor de spanningen over de andere twee weerstanden geldt:

$$\begin{aligned} U_{R2} &= 27k\Omega * (I_b + 70\mu A) \\ U_{R1} &= 70\mu A \end{aligned}$$

Alle spanningen bij elkaar opgeteld moeten 9,6V zijn (de voeding), dus:

$$\begin{aligned} 9.6V &= U_{R3} + U_{R2} + U_{R1} \\ &= 1k\Omega * 101 * I_b + 0.07V \\ &\quad + 27k\Omega * I_b + 1.89V + 0.7V \\ 7V &= 101k\Omega * I_b + 27k\Omega * I_b \\ &= 128k\Omega * I_b \Rightarrow I_b = 55\mu A \end{aligned}$$

$$I_c = h_{FE} * I_b = 100 * 55\mu A = 5.5mA$$

$$U_{R2} = (27k\Omega * (55\mu A + 70\mu A) = 3.375V$$
$$U_{R1} + U_{R2} = 0.7V + 3.375V = 4.075V$$

En dat is heel wat meer dan de 2.59V die ik even snel berekend had. Voor het doel waar het

toe diende, even snel de ordegrootte van de spanning bepalen, was het goed genoeg, maar voor de berekening van de versterker zou het niet goed genoeg geweest zijn. Zo zie je maar, je kunt niet altijd zomaar de basisstroom verwaarlozen. Paul bedankt voor de aanvulling!

Chinese componenten

We kopen allemaal wel eens iets bij AliExpress of soortgelijke goedkope Chinese leveranciers. Dat dat niet altijd een goed idee is, bleek uit de ervaring van een van de nabouwers van mijn L-C-R-ESR meter. Ik ontving een e-mail van Rolf Snijder die problemen had met de meter: de uitlezing was niet stabiel te krijgen en vloog alle kanten op. Mijn eerste advies was om de voeding wat beter te ontkoppelen: ik vermoedde een goedkope stekkernetvoeding die wellicht problemen gaf. Maar dat bleek niet het geval: het leek wel iets beter te worden na het ontkoppelen, maar het probleem was zeker nog niet opgelost. Mijn tweede vermoeden was het 74HC132 IC, en nog voor ik dat aan Rolf kon voorstellen, kreeg ik een e-mail waarin hij meldde dat het probleem was opgelost. Hij had 10 van die 74HC132's in China besteld, maar die gaven dus inderdaad problemen. In de meter worden ze gebruikt als vrijlopende oscillator op ongeveer 100kHz en daar bleken ze niet erg stabiel in te zijn. Rolf kocht er 4 bij een Europese firma en ziedaar: probleem opgelost. Dus: Ja, Chinees spul is goedkoop. Maar vooral van elektronica componenten weten we dat die vaak slecht nagemaakt en van erbarmelijke kwaliteit zijn. Let dus op waarvoor je goedkope componenten toepast: ze doen het niet overal even goed...



Afdelingsnieuws

Juni is de laatste maand voor de zomerstop en daarin komen we nog twee keer bij elkaar: op de woensdagen 12 en 26 juni. In de maanden juli en augustus zijn er geen bijeenkomsten en pas op 11 september beginnen we weer met het nieuwe seizoen. Dat betekent dat 12 juni voorlopig de laatste kans is om QSL kaarten uit te wisselen, daarna moet je 3 maanden wachten! Zorg dus dat je je kaarten nog inlevert op 12 juni. Dat er geen bijeenkomsten meer zijn in juli en augustus wil niet zeggen dat we niets meer doen. Aangezien er voldoende belangstelling voor de iGate is, zullen we deze weer eenmalig als kit aanbieden. Daarvoor moet nog de exacte prijs vastgesteld worden en dat is het eerste wat we nu zullen regelen. Vervolgens komt weer de gelegenheid om definitief in te tekenen op onze website. De timing is wat dat betreft wel ongelukkig, omdat we nu het vakantieseizoen tegemoet gaan. Dan lopen we een kans dat geïnteresseerden het project mislopen. Weet je nu al heel zeker dat je een iGate wil bouwen, stuur dan even een mailtje naar info@pi4raz.nl. Mochten we je aanmelding dan niet gezien hebben, dan nemen we in elk geval contact met je op om te zien of je de inschrijving niet gemist hebt. Zodra we de definitieve prijs van de iGate berekend hebben, zullen we de inschrijving openen. Let daarvoor op onze website en/of Facebook pagina.